NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ONBOARD AND PARTS-BASED OBJECT DETECTION FROM AERIAL IMAGERY

by

Robert Michael Zaborowski

September 2011

Thesis Co-Advisors: Mathias Kölsch
Chris Darken

This thesis was done at the MOVES Institute.
Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
23–9–2011	Master's Thesis	2010-04-28—2011-09-20

4. TITLE AND SUBTITLE

Onboard and Parts-based Object Detection from Aerial Imagery

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Robert Michael Zaborowski

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Department of the Navy

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number XXX.

14. ABSTRACT

The almost endless amount of full-motion video (FMV) data collected by Unmanned Aerial Vehicles (UAV) and similar sources presents mounting challenges to human analysts, particularly to their sustained attention to detail despite the monotony of continuous review. This digital deluge of raw imagery also places unsustainable loads on the limited resource of network bandwidth. Automated analysis onboard the UAV allows transmitting only pertinent portions of the imagery, reducing bandwidth usage and mitigating operator fatigue. Further, target detection and tracking information that is immediately available to the UAV facilitates more autonomous operations, with reduced communication needs to the ground station. Experimental results proved the utility of our onboard detection system a) through bandwidth reduction by two orders of magnitude and b) through reduced operator workload.

Additionally, a novel parts-based detection method was developed. A whole-object detector is not well suited for deformable and articulated objects, and susceptible to failure due to partial occlusions. Parts detection with a subsequent structural model overcomes these difficulties, is potentially more computationally efficient (smaller resource footprint and able to be decomposed into a hierarchy), and permits reuse for multiple object types. Our parts-based vehicle detector achieved detection accuracy comparable to whole-object detection, yet exhibiting said advantages.

15. SUBJECT TERMS

Onboard Object Detection, Parts Based Object Detection, Computer Vision, Machine Learning

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	73	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

ONBOARD AND PARTS-BASED OBJECT DETECTION FROM AERIAL IMAGERY

Robert Michael Zaborowski
Lieutenant, USN
B.S., United States Naval Academy, 2004
M.E.M., Old Dominion University, 2010

Submitted in partial fulfillment of the
requirements for the degrees of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

and

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2011

Author:	Robert Michael Zaborowski Lieutenant, USN
Approved by:	Mathias Kölsch Thesis Co-Advisor
	Chris Darken Thesis Co-Advisor
	Mathias Kölsch Academic Associate, MOVES
	Peter J. Denning Chair, Computer Science Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The almost endless amount of full-motion video (FMV) data collected by Unmanned Aerial Vehicles (UAV) and similar sources presents mounting challenges to human analysts, particularly to their sustained attention to detail despite the monotony of continuous review. This digital deluge of raw imagery also places unsustainable loads on the limited resource of network bandwidth. Automated analysis onboard the UAV allows transmitting only pertinent portions of the imagery, reducing bandwidth usage and mitigating operator fatigue. Further, target detection and tracking information that is immediately available to the UAV facilitates more autonomous operations, with reduced communication needs to the ground station. Experimental results proved the utility of our onboard detection system a) through bandwidth reduction by two orders of magnitude and b) through reduced operator workload.

Additionally, a novel parts-based detection method was developed. A whole-object detector is not well suited for deformable and articulated objects, and susceptible to failure due to partial occlusions. Parts detection with a subsequent structural model overcomes these difficulties, is potentially more computationally efficient (smaller resource footprint and able to be decomposed into a hierarchy), and permits reuse for multiple object types. Our parts-based vehicle detector achieved detection accuracy comparable to whole-object detection, yet exhibiting said advantages.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Introduction **1**
1.1 Operational Need 1
1.2 Research Questions 2
1.3 Organization of Thesis 3

2 Related Work **5**
2.1 Features, Descriptors and Detectors 5
2.2 Object Detection Methods 5
2.3 Part-based Training 7
2.4 Part-based Classifiers Without Structural Models 8
2.5 Part-based Classifiers Using Structural Models 9
2.6 Obstacles to Object Detection 12
2.7 Detection from Aerial Imagery 13

3 Methodology **15**
3.1 Vehicle Detection in Aerial Imagery 15
3.2 Onboard Detection. 16
3.3 Object Rotation 17
3.4 Parts-based Detection 18

4 Experimentation **23**
4.1 Vehicle Detection in Aerial Imagery 23
4.2 Onboard Detection. 24
4.3 Accounting for In-plane Object Rotation 24
4.4 Parts-based Detection 25

5 Results **29**
5.1 Vehicle Detection in Aerial Imagery 29
5.2 On-board Detection 31

5.3 Accounting for In-plane Object Rotation 32
5.4 Parts-based Detection 33

6 Discussion **39**
6.1 Vehicle Detection in Aerial Imagery 39
6.2 Onboard Processing 40
6.3 Accounting for In-Plane Object Rotation 40
6.4 Parts-based Detection 41

7 Conclusions **43**
7.1 Detection of Vehicles in Aerial Imagery 43
7.2 Onboard Object Detection 43
7.3 Accounting for In-plane Rotation 44
7.4 Parts-based Detection 44
7.5 Operational Implementation 45

List of References **49**

Appendices **51**

A Appendix A **53**
A.1 Decision Tree 53
A.2 SVM 53
A.3 Bi-directional Kohonen Self-Organizing Map 53
A.4 Neural Network 54
A.5 Parts-based Detector ROC Curve 54

List of Figures

Figure 3.1 A sample test image, pos24my001_010. 19

Figure 3.2 The discrete detections for a 30 stage rear driver's side corner detector of test image pos24my001_010. 19

Figure 3.3 The scored detections for a 30 stage rear driver's side corner detector of test image pos24my001_010. 19

Figure 3.4 Graphical illustration how the four corner detectors are applied to an image. 21

Figure 3.5 Illustration of how the discrete map sets are generated for a sample. . . 21

Figure 3.6 Two examples of features defined by a pair of part detections. 22

Figure 4.1 Picture of the Sig Rascal 110 ARF remote controlled aircraft. 23

Figure 4.2 This shows the training set as it is provided to the OpenCV haar cascade training algorithm for the non-rotated cascade. 25

Figure 4.3 This shows the training set as it is provided to the OpenCV haar cascade training algorithm for the non-rotated cascade. 26

Figure 4.4 This shows the subset of the training set for the rotations applied to the first training image. 27

Figure 4.5 This shows the subset of the training set for the rotations applied to the second training image. 28

Figure 5.1 Graph of available network bandwidth over wave relay at Camp Roberts. 31

Figure 5.2 The graph above shows the performance of the detector trained with an aligned positive imagery set. 33

Figure 5.3 The graph above shows the performance of the detector trained with a rotated positive imagery set. 34

Figure 5.4 Graphical illustration of the trained decision tree model. 35

Figure 5.5 Graphical illustration of the trained Bi-Directional Kohonen Self-Organizing Map model. 37

Figure 5.6 The ROC curve for the parts-based detector compared to the aligned Cascade 02 detector, on a logarithmic scale. 38

Figure A.1 The ROC curve for the parts-based detector compared to the aligned Cascade 02 detector, on a logarithmic scale. 55

List of Tables

Table 5.1 Detector detection performance during testing at Camp Roberts in conjunction with TNT 10-4. 29

Table 5.2 Detector speed performance during testing at Camp Roberts in conjunction with TNT 10-4. 30

Table 5.3 Detector performance during testing at Camp Roberts in conjunction with TNT 11-3. 32

Table 5.4 Detector speed performance during testing at Camp Roberts in conjunction with TNT 11-3. 32

Table 5.5 Confusion matrix showing the results of the decision tree model. 34

Table 5.6 Confusion matrix showing the results of the SVM model. 36

Table 5.7 Confusion matrix showing the results of the Bi-Directional Kohonen Self-Organizing Map model. 36

Table 5.8 Confusion matrix showing the results of the Neural Network with 3 units in the hidden layer. 36

Table A.1 Confusion matrix showing the results of the decision tree model applied to the training data. 53

Table A.2 Confusion matrix showing the results of the SVM model applied to the training data. 53

Table A.3 Confusion matrix showing the results of the Bi-Directional Kohonen Self-Organizing Map model applied to the training data. 54

Table A.4 Confusion matrix showing the results of the Neural Network with 2 units in the hidden layer applied to the training data. 54

Table A.5 Confusion matrix showing the results of the Neural Network with 2 units in the hidden layer applied to the test data. 54

Table A.6 Confusion matrix showing the results of the Neural Network with 3 units in the hidden layer applied to the training data. 55

Acknowledgements

Thank you to my parents for their encouragement and faith that I would be able to finish this thesis.

I would like to thank Mathias Kölsch for his encouragement and patience that enabled me to complete this thesis. Thank you for your constant support, available even at odd hours of the day and weekend.

To Chris Darken, thank for your providing a sounding board and asking thought provoking questions about this thesis so that I would deliver a good final product.

To Vladimir Dobrokhodov, Kevin Jones, Tim Chung, and Jeff Wurz, thank you for adapting members of the Rascal fleet and associated software for the TNT experiments. It was a very rewarding experience to see my efforts actually produce a proof of concept in a real world exercise.

To Justin Jones, my friend and college who helped me get started with OpenCV, annotating samples, and building detectors.

Thank you to Space and Naval Warfare Systems Center, Pacific, especially my mentor Daniel Cunningham, whose fellowship grant enable the purchasing of hardware to support the field experiments in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

Rapid planning and decision making has become a focus for future development of the armed services. The rate of UAV procurement has increased significantly in recent years, and increased the availability of UAV assets for intelligence gathering. Imagery is one sensor that UAVs are capable of carrying, but the amount of raw data collect creates a deluge that is beyond the available capacity of human analysts.

The result is that operational resources are utilized to perform data collection without the ability to perform that analysis to capitalize on information available in the data collection. Flying a UAV for data collection also puts the UAV at risk to damage or loss, which can adversely affect the ability to perform data collections in the future.

This thesis provides supporting evidence that computer vision algorithms can support a workload reduction for analysts, immediate feedback, retains collection data onboard the UAV for other services, and reduces the network bandwidth usage for relaying high quality still images.

Parts-based detection, using an adaptable structure model, would allow for the detection of rigid and articulated objects. A parts-based detector uses several small detectors for each part. The observed presence of absence of these parts is an intermediate feature set. This intermediate feature set is used by a structural model to determine if the whole object is present or absent. This thesis implemented a structural model that was an adaboost decision tree taking observation maps, one for each part, and determining the presence or absence of an object.

1.1 Operational Need

Given that there is an insufficient pool of human analysts to perform a single review of collected imagery, seldom can a second review be performed to ensure all possible information was gleaned from the data collection.

The result is that operational resources are utilized to perform data collection without the ability to perform that analysis to capitalize on information available in the data collection. Flying a

UAV for data collection also puts to UAV at risk to damage or loss, which can adversely affect the ability to perform data collections in the future.

1.1.1 Increased Situational Awareness

Processing of imagery while in flight allows for the UAV to immediately provide the detections to the controlling station. This provides real-time feedback to the operator the time and location of a detection. Knowing the time and location of a detection can allow operators to focus their efforts in a particular geographic area for the remainder of the flight. Focusing on a higher contact area can provide more relevant data collection and increase the utility of reconnaissance flights. After the UAV lands and the full high resolution images are downloaded, analysts can use the detections as indications of which images should be reviewed for additional context, and/or which images should be checked for missed detections.

1.1.2 Increased Autonomy

UAVs are currently remote controlled and have little autonomy. If collected imagery is analysed by the ground station, the collected information must be sent back to the UAV for it to take action. By performing data analysis onboard the information gained from the detections is available to other processes onboard the UAV. Automation of any level will require inputs of this type if a UAV is to react to object detections. Real-time object detection onboard the UAV can quickly provide the UAV with data points from which to make decisions.

1.2 Research Questions

This thesis addresses the following research questions:

(a) Can vehicles be detected in aerial imagery?

(b) Is it possible to perform vehicle detection on-board the UAV and provide results to a controlling ground station in a timely manner?

(c) Is it possible to use an automated or semi-automated algorithm for training a structural model that is robust to a single point of failure and uses an intermediate feature set for object detection?

Viola-Jones classifiers were trained and implemented in a laboratory environment and field exercises to show real-world capabilities of the computer vision algorithms.

1.3 Organization of Thesis

This thesis is organized as follows:

(a) Chapter 1 discusses the contribution of this thesis, and potential operation role of object detection.

(b) Chapter 2 addresses related work.

(c) Chapter 3 discusses the methodology used for this thesis.

(d) Chapter 4 details the experiments and how they were conducted.

(e) Chapter 5 reports the results of the experiments performed.

(f) Chapter 6 discusses the results.

(g) Chapter 7 provides conclusions that can be drawn from the work performed in this thesis.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Related Work

2.1 Features, Descriptors and Detectors

Features form the building blocks of object detectors that include scale invariant feature transform (SIFT) [1], speeded up robust features (SURF) [2], and Viola-Jones [3]. However robust the detector even good features can become occluded [4].

2.2 Object Detection Methods

There are a variety of ways to use a single image, or multiple images taken within short time periods to identify an object's location or absence in an image or sequence of images. Where the whole object detector looks for the entire object in one sweep, it may only find the object in a particular pose, while the part-based detector finds the object in multiple poses. Lighting also can be an obstacle to object detection, which is a strength of the 3-D wireframe method that attempts to account for lighting differences. Some objects may be in motion, which can be detectable over a series of sequential images. Various techniques have their respective strengths and weaknesses, such that at this time there is no one best way, but instead several choices that when applied to a particular object in certain environments can produce improved recall and a reduced false alarm rate.

2.2.1 Whole Object Detection

Appearance-based or shape-based detection can be conducted for an entire object in one step. The more variations in the object's appearance, the more generic the detector must be made. Depending on the feature type used in the whole object detector may be robust to object rotation [2]. A cascaded detector, which has multiple stages each composed of several weak classifiers, can improve object discrimination while maintaining recall performance. The Viola-Jones cascaded detector can reduce the number of features evaluated by applying a threshold value to each stage of the detector before continuing on to the next stage [3]. It is also the case that for articulated objects several detectors may be required to account for the variations in pose that the articulated object can present, because the training data is aligned to one uniform orientation to allow for optimal feature extraction.

2.2.2 Detections Over Time

Imagery collected over a time period can be analysed for similar features of detection by hypothesizing the probable translation of the detected features and verifying this translation across the feature locations [5, 6, 4]. To use time as an additional dimension the camera must either be fixed [7], have a short temporal interval between overlapping frames, or the vision software must have the ability to register similar features in the overlapping images while detecting object motion [8]. Tracking of multiple objects may also be limited if the objects move within a close proximity of each other and one or more objects is partially occluded by another, and when an object enters or exits the camera's field of view [7].

2.2.3 3D Wireframe Model Detection

3D wireframe models are constructed from the structural description of a vehicle, which is useful in accounting for variances that can require very meticulous composition of the training data. Where a 2D appearance based model relies on the training data to learn variances in appearance, the 3D model is able to use the structure of the vehicle to determine the variance of an object, for example a vehicle's shadow. A 3D model can be used to determine where the vehicles shadow would appear for a given light source, as well as the effects this light source would have on the vehicle's surfaces. One detection implementation extracts edges pixels and computes gradient direction which are used to determine lighting effects. Using the calculated lighting effects to account for variances in the image the edge features can be compared to the 3D model likelihoods to produce a posteriori and a combined matching score for the object [9].

2.2.4 Part-based Detection

While rigid objects will retain their same shape, articulated objects are composed of the same parts but arranged differently. Articulated objects can be detected by making observations for their composite parts and building up to the whole object. Acquiring parts from a set of sample images that depict the object for which a detector is being built can account for the variability that exists across different objects belonging to the same class [10]. A part can be described as a shape, which can be defined by a group of key points or edges [11]. The collection of key points and edges that define a part are less likely to occur than the individual key points and edges that make up the part, which means that searching for parts will yield less false positives [12]. Handling parts that represent common boundary and edge structure would allow for the sharing of parts across classes in the future. By sharing common parts across classes, the same detectors can be reused with different structural models, such that related and non-related

object classes can be detected by a single detector using the same parts codebook. However, the commonality of smaller, more basic parts require that the relative spatial arrangement of the parts be modelled to selectively prune the large number of background detections [13].

2.3 Part-based Training

Once the parts are selected and the detectors are trained, the part detections must be combined in some way. Without a defined structure that relates some number of the parts to each other spatially or geometrically, we can rely on the number of detections and possibly confidence levels — if the detections are capable of returning probabilities of that part's existence. With a defined structure we can eliminate object detections in the event the parts do not conform to the specified model, because if the detections do not conform to the specified model then they are not arranged correctly for the object class we are searching the visual data for. A structure can be determined within the same range as the parts were selected, fully manual to fully automated.

2.3.1 Part Selection

The parts that are used to describe an object must be selected, which can be done manually or automatically. A fully manual method requires a human operator to select each part from the training image set, whereas a fully automated solution would take into account the training images and select the optimal parts and find those parts in all images. These two methods are the extremes of the range of part selection and training solutions, and a variety of methods exist between these extremes. The automated extreme requires that high quality images of the object for which a detector is to be built be provided, from which the chosen algorithm will extract parts to be used for object detection.

Supervised Part Selection

Supervised part selection requires a database of labelled parts, which must be created if it does not already exist. This is a time intensive process, that also requires the human labelling to be consistent throughout the database. While a higher level of supervision may improve classifier performance, it can also degrade performance if suboptimal parts are chosen [14]. These trends suggest that automated labelling can improve performance by allowing the system to find the parts it determines to be most discriminative, while reducing the time spent labelling training sets [15].

Unsupervised Part Selection

Annotating parts is a very time consuming process that also hinges on the annotator's ability to determine the correct parts and ensure all annotations are similar. Using an unsupervised method to determine which parts should be used and their relationship(s) can save annotation time, in exchange for computational time. Using real AdaBoost to develop a tree structure based on weak classifiers, which collects data and splits when the false alarm rate exceeds a threshold [4].

2.4 Part-based Classifiers Without Structural Models

While no spatial information may be used to learn a physical structure for an object or object classes, there is still a requirement to learn how feature or part detections can be used to discriminate between classes. It is possible to constructor a detector that uses only the detection of parts to determine the presence of the entire object, without the use of any structure [16, 17]. One example is the use of Viola-Jones like cascaded classifiers to find parts and weight each part's detection. The sum of the weighted part detections compared to a threshold then determines if the whole object is present [16]. A bag-of-words model can make use of a visual descriptor and perform unsupervised learning on a collection of images to statistically model the occurrence of visual features in a class of objects, which has been demonstrated using SIFT descriptors [17, 18]. The bag-of-words model (also known as a bag-of-features model) uses a compact histogram representation for image classification based on observed and unordered appearance descriptors. The use of an unsupervised learning technique reduces the time intensive annotating process and thereby increases the number of available training images to all high quality images representative of the object I would be looking for. The bag-of-features model uses histograms of appearance features to classify objects. Codebook generation of this histogram can be automated by performing K-means clustering on extracted descriptors [18]. Global part detections may exist in any location of the object and provide no spatial information about the part [19]. Features that are characterized by the global shape of an object are susceptible to noise and background clutter [20].

Given the orientation and location of edges within a specified window, the most likely part can be calculated from a mixture model. Once the most likely parts have been determined, a different mixture model is used to determine the object class based on the occurrence of the observed parts, without use of the part's spatial information [12].

2.5 Part-based Classifiers Using Structural Models

The structure makes use of information that may be geometric or spatial in nature to recombine part detections and determine if the whole object is present. Local features are constructed from a feature detector and a feature descriptor, and can be applied in a three step or four step process. The three step process is comprised of: feature detection, feature description, and feature matching. The four step process, of which bag-of-features would be a good example, also uses feature detection and feature description but then goes on to cluster the features and finally constructs a frequency histogram [21]. Local part detections express spatial relationships to the object [19], and allow for incorporation of orientation, gradient, and probabilistic co-occurrence into a model to reduce false positives, and increase recall [22]. Features based on the observation of a part's presence, or the relationship between the observations of the presence of multiple parts, are known as structural descriptors. Structural descriptors composed from local part detections are more robust to noise and background clutter, compared to structural descriptors composed from global part detections [20].

2.5.1 Voting Maps

Determination of a rigid object's center from at least three points is possible with local part detections, where the part detections will cast votes for object center and scale based on the codebook entries [11, 23]. A codebook of possible variations in the object's appearance is based on the parts, or detected primitives that store location, scale, elongation, and rotation information. To account for initially missed parts in a bottom-up approach each part can be model using naive-Bayes, assuming independence for each part detection. From the maximized response of the individual part detectors a top-down approach can be used to estimate the location of missing parts. Using the probability estimates of the parts recovered in the top-down phase for calculation of the maximized weighted log-likelihood of the observed object pose provides better results than thresholding only those parts detected in the bottom-up detection phase [23].

2.5.2 Spatial Models

Spatial models use the relative distance and position information between multiple parts to identify the structure of the whole object. These models may add additional information to increase the robustness of the model. Geometric features, differences in size, orientation, height and width ratios between part detections, are an example of additional information that can be used to determine if two discrete parts are detections belonging to the same whole object.

Spatial Only Models

Use of a Canny-Edge-Detector and Harris Detector for edges and corners paired with their associated Mahalanobis distance between these identified points of interest is effective for rigidly formed objects such as vehicles. Kaaniche et al. have shown that corners are repeated throughout a sequence of images with a higher probability than do edges, indicating that corners are a better part to detect than edges of a vehicle [5].

Exploitation of spatial relationships between parts can be used to discriminate false part detections [10, 24, 25, 26]. In the work of [25] a codebook is created for each object, and each entry in the codebook has an associated spatial distribution used during the voting process to determine an object's center. Direction can be paired with the distance for a more precise spatial model, and in the work performed by Agarwal et al. eight 45° ranges define the possible direction descriptions between parts. This specific method considers parts in a fixed order, reducing the number of direction bins necessary for the model to consider, reducing space and computational requirements. Classifications are binary and no probability model is used. A whole object detection is based on the presence or absence of specific parts, for specific orientations and spatial relationships of the parts [10]. Alternatively direction can be described by separating out the x-component and y-component distances if the aspect is known and rotation can be controlled [26].

Generative models can be used to discover specific features within a larger object, a top-down part localization approach. Once the larger object is detected it can be searched for parts of interested by collecting spatial information such as distance between the parts, and ratios of the distances between groups of parts. These detections and their spatial information, available due to the use of Haar-like wavelet features, can become the input for training a likelihood ratio model [24].

Global and Spatial Models

The fusion of global and local part detections can increase recall if some part detections are absent, while increasing the level of discrimination to reduce false positives. Spatial relationships between local features can be further developed into chains of parts that exist with a spatial relationship to each other if the object of interest is present [19].

Normalized Spatial Models

Combining a small set of geometric features with spatial features between parts of an object can provide accurate structural descriptions of objects that are in-plane rotation and scale invariant. Shapes are defined by line segments or ellipsoids which are segmented based on color, and receive a quality score based on the line segments' angles of intersection. Geometric features for this particular detector included ratios of length and width between neighbouring shape and length to width ratios of the two neighbouring shapes. The spatial relationships are defined from points on the line edges of the two neighbouring parts shapes. This technique has been shown to be robust to out-of-plane rotations of up to 45° [20].

2.5.3 Homogeneous Graphical Structures

Graphical structures can use spatial information, and that spatial information can relate objects that are next to each other or across the image [27, 28, 29, 14, 30]. Making use of spatial relationships between parts within a graphical structure can allows for exclusion of invalid whole object detections based on local or long-range interactions [29]. While a large number of parts can lead to intractable problems, graphical structures enable a rigorous probabilistic analysis of the problem, and avoid the explicit enumeration of the full set of hypotheses by limiting the direct dependencies. Part detections are used to develop a patch layer that can propagate belief messages between nodes. Messages are sent using biggest-first scheduling. The result of using the messages to update the class probability of each node will cause a global convergence of the graph to the object class present in the image [27].

In the work [13] edge boundaries are used to detect parts, each of which has its own trained detector which can produce multiple returns when applied to an image. For every part a probability distribution is used to define probable locations of other parts for every detection. Using probability distributions reduces the search space to plausible values, however there is an implicit assumption that all parts are detected by the part detectors. If a part is missed by the detector due to a failure to identify it or occlusion the algorithm fails and the object will not be detected. For each configuration a score is generated for the possible transformation which would result in the spatial configuration being investigated, and if this score is found to be consistent with the calculated prior an elliptical region is generated to identify the predicted object location in the image [13].

Using two codebooks, one of car parts and another of background objects, and a directed graph,

vehicle detections can be made in images that contain clutter and partial occlusion. The image is blurred to remove surface markings during training, and k-means is used to determine the median value of the object's surface areas, which are stored in the codebook. This method makes use of part-part and part-object linkages to correctly identify the object's class as vehicle or background object. Due to the high dimensionality of the possible solution space Markov Chain Monte Carlo (MCMC) is used as an approximate inference method, but incurs a high cost in terms of the number of samples taken and therefore time to solution. To speed to MCMC inference Metropolis-Hastings (M-H) is used to to provide a general approach for producing a bottom-up inference [28].

2.5.4 Heterogeneous Graphical Structures

The star-graph model has a root node from which the spatial relationships to all other parts are measured. Whole object detection is conditioned on the detection and spatial relationship between the root node and all other nodes [14]. An extension to the star-graph model is to create a set of root node parts, which increases the representational power at the expense of increased computational cost. The *k*-fans structure seeks to allow for a balance between the representational power and computational power by allowing the user to specify the number of nodes(k) [30].

2.6 Obstacles to Object Detection

Illumination, rotation, and different viewpoints of objects from the same class changes the appearance of similar objects between the different sequences of imagery, which must be accounted for by the codebook [11, 15]. While more complicated models can be more exacting in detail and more expressive over a wider range of variations for a class of object, they are historically outperformed by more simplistic models. More robust and complicated models suffer from the increased difficulties of training a more complicated model [14].

2.6.1 Developing Scale Invariance

Given a single scale detector the image can be rescaled over a specified range or scale factors at specified intervals or magnitudes. The single scale detector is then run over the resized image to detect objects of interest within range of scales the detector has been trained to classify [10]. Depending on the probable size of the detection different parts may or may not be recognizable, such that using multiple detectors that work at different scales can be used to improve detection performance [31].

2.7 Detection from Aerial Imagery

Computer vision systems for UAVs have the added challenge of accounting for the suboptimal images that UAVs can produce. The faults and artifacts in the images are caused by the motion of the UAV on multiple axes, which results in images with a perspective other than near nadir or outside of the desired slant angle's threshold, in addition to the barrel distortion of the camera. To correct for the irregularities caused by the UAV's motion a preprocessing step must occur that requires input from on-board sensors including pitch, roll, and yaw, each of which have their own error. Performing this preprocessing step can reduce the need for human-in-the-loop oversight of the computer vision by providing better raw data for the computer vision system to interrogate, which provides more accurate output [32].

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 3:
Methodology

This chapter details the challenges of transitioning from ground station processing to onboard processing, the part-based detection method, and how the detectors were built. For the purposes of experimentation and testing vehicles were selected as object of interest, although Viola-Jones detectors and parts-based solutions similar to the methods described in the thesis have been applied to other objects, such as an AK-47 [33].

3.1 Vehicle Detection in Aerial Imagery

Currently, there exists no commercially available software capable of detecting vehicles in aerial still imagery. Most detectors currently in use rely on change detection over time, for example motion detection, which requires full motion video. There are several detection methods capable of detecting faces, for example, so the construction of a vehicle detector from the near nadir viewing aspect was hypothetically possible. For this thesis, several Viola-Jones detectors were built and tested. One of those detectors was selected for use at the Naval Postgraduate School Tactical Network Topology (TNT) exercises in August, 2010 and June, 2011. The purpose of testing was to evaluate the usefulness of the trained detector and the level of performance that could be achieved in terms of recall, false positive rate, and processing time.

The first objective was to determine if vehicle detection in aerial imagery was possible. Transmission of the entire raw image to a ground-based workstation for processing was selected as a proof of concept to avoid processing restrictions due to limited resources encountered in onboard UAV systems.

At TNT the Rascal UAV collected sample imagery by flying overhead and orienting the camera at a near nadir aspect. At the time of this experiment no vehicle detector had been developed for this application. The Viola-Jones detector was selected as a state-of-the-art detection method for vehicle detection based on its proven performance with rigid objects (reference). To train a Viola-Jones detector positive and negative image sets are required for the training algorithm to select and evaluate appropriate features. The negative set of training images should be representative of the likely background where detection will be attempted. The positive set of images should be representative of the objects as they will most likely exist in any test images. The

detectors trained for this thesis used positive and negative images that were annotated from previously collected Rascal UAV imagery that contained vehicles in the near nadir aspect.

Once the proof of concept had been successfully demonstrated, the transition from object detection on a desktop to object detection onboard the UAV could be undertaken. The computational power available on the UAV would be sparse compared to the desktop, which meant that system integration would be more difficult, and that the algorithms needed to be finely tuned to minimize resource utilization. Tuning of detector parameters was performed during testing with detection performed on the desktop, but actual integration of the vehicle detection onboard the UAV was more complicated.

3.2 Onboard Detection

Implementing the detection process on the UAV had the potential to reduce network bandwidth usage, compared to transmitting entire raw imagery. To demonstrate the potential bandwidth reduction by performing object detection on-board, the existing Rascal UAV's payload was modified as necessary to test onboard detection. By performing the detection onboard the UAV, the detections could be cropped out from the original full image, which would allow the UAV to reduce bandwidth consumption by transmitting the smaller detections as cropped images with location meta-data.

3.2.1 Onboard Resources

Although written to be portable, implementing the existing detection algorithms onboard the UAV still required extensive integration. Achieving real-time processing with the limited resources available on the UAV required the introduction of configuration inputs that limited the range of scales over which the detector would search. These configurations inputs and limits were imposed by code modifications to the custom code produced for this thesis, and modifications to functions in the OpenCV library. The existing PC-104 was not adequate for on-board processing in real-time. By selecting a newer, more powerful PC-104 real-time processing would be possible, but the power consumption necessitated a transition from two PC-104 boards to the single, more powerful board. This meant that flight and payload operations would be conducted on the same PC-104, so the object detection algorithm would not have sole claim to all processing power. To allow the flight critical systems access the PC-104 board as necessary, the vehicle detection program was run in at a lower process priority, "nice."

3.3 Object Rotation

The Viola-Jones cascade detector uses haar-like features and is not inherently rotation-invariant, like SIFT or SURF features.

Training a detector requires sets of both positive and negative training images. The negative set of training images should be representative of the likely background where detection will be attempted. The positive set of images should be representative of the objects as they will most likely exist in any test images. A common obstacle is rotational invariance, because the positive images are typically aligned in the same direction to allow for the most representative features to be extracted by the training software. For this detector this means that all cars would be facing the same direction. Two methods of overcoming this obstacle are examined, the faster of the two was used for the ground station and on-board processing experiments.

3.3.1 Aligned Training Set

A training set with all images aligned to the same orientation creates a detector for the object in the positive images, for the particular direction of the object in the positive images. Rotational variance in the test set theoretically produces poor performance. The degree to which the performance decreases hypothetically becomes worse as the variance in orientation increases. To overcome this limitation of the Viola-Jones detector, the test image is rotated through the entire 0 to 360 degree rotational space in a specified increment. This process requires that the image be rotated in real-time and the detector be applied after each rotation.

3.3.2 Rotated Training Set

Rotating the training set in at specified intervals of the 0° to 360° in-plane rotational space will provide the training algorithm all possible variances of in-plane rotation from which to develop a rotationally invariant classifier. If a sufficient level of robustness to rotational variance is achieved, running this single detector over an image can replace the need to rotate an image for the aligned detector. Removing the need to rotate the image at run-time also means that artifacts from pixel interpolation will not be present.

To test detector sensitivity to rotation two detectors where trained, one using an aligned positive training set, and one using a rotated negative training set. Both detectors are tested against rotated positive image sets to develop an ROC curve for each rotational increment. Comparing

the variance of the ROC curves graphically revealed the significant rotational sensitivity of the aligned training set detector, compare to the rotationally robust rotated detector.

3.4 Parts-based Detection

3.4.1 Part Selection

The four vehicle corners were selected for as the salient parts for this experiment. Selection of the corners as descriptive parts of a vehicle was a decision made by human operators. To validate the descriptive power of corners for vehicle detection one corner detector were trained and tested for each corner of a vehicle.

3.4.2 Choice of Feature Descriptors

While vehicle corners are essentially edges in two directions, using solely an edge detector would limit future parts in a parts-based model to other corners present in the vehicle. Using a Viola-Jones cascade with Haar-like features allows for training of larger variety of parts using the same techniques that will be developed for the vehicle corners.

3.4.3 Part Annotation Technique

Rotating all images from which the parts were collected to a standard pose, hood facing left, the corners were then known to be at the same relative positions in all images for which training was to be conducted. This reduced annotation time by automating corners selection based on image geometry.

3.4.4 Part Detections

The Viola-Jones detector searches an image using sub-windows, regions of the image that are the size of the detector. Each stage of the detector has a set of features and a threshold value. The windowing processes steps through all window in the image, applying the weak classifiers for until a stage fails or all stages successfully pass the threshold value.

Discrete Detections

All detectors are applied at each possible location of a test image, across all image scaling values from 1.0 until the width or height are equal to the size of the detector. All scales are searched using the cascade approach present by Viola-Jones. The detector is limited to a boolean return if all cascade stages in the selected detector are successful. Figure 3.1 is an example test image, and Figure 3.2 is the discrete detection map of a 30 stage rear driver's side corner detector run on the test image. In Figure 3.2 the two white pixels are the discrete detections. To graphically

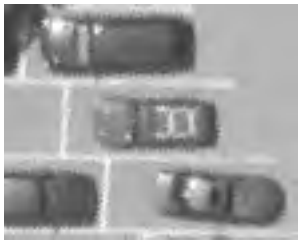

Figure 3.1: A sample test image, pos24my001_010.

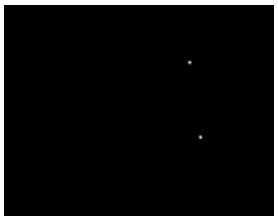

Figure 3.2: The discrete detections for a 30 stage rear driver's side corner detector of test image pos24my001_010.

display the detection maps, the value of 0.0 to 1.0 is scaled over the 256 possible grayscale values. A discrete detection has a value of 1.0, so the grayscale value in the image is 255.

Scored Detections

Scored detections can be generated by dividing the number of stages passed by the number of cascade stages used by the detector. Performing this for each pixel will create a map of scored observations. This is useful when dealing with partial occlusion such that some features of a part are present, but not all. With absent features the detector should pass some of the cascades, but most likely will not pass all cascades. Scoring the observations allows for the available data, the features that are present, to be combined with data available from the other part detectors to create a final composite map of an object's presence or absence from specific pixels of the original image. Figure 3.3 is an example of the scored detection map run over the test image in Figure 3.1. Figure 3.3 is scaled the same way as Figure 3.3, but has many more pixels of

Figure 3.3: The scored detections for a 30 stage rear driver's side corner detector of test image pos24my001_010.

varying intensity white because it shows the pixels that did not pass all stages of the cascade.

3.4.5 Structural Model

The structural model is developed in this thesis uses the detection maps from the part detectors. Each part detector will return a scored detection map the size of the original image, minus the dimensions of the detector. For example, a detector width A and height B, run over an image of width C and height D will return a map of width $C - A$ and height $D - B$.

Training the Structural Model

Individual part detectors must be trained before the structural model can be trained. The size of the structural model, in pixels, is specified when training commences. The automatic part annotation algorithm uses a ratio method to extract the part samples from cropped positive samples for training. Using this ratio means that the size of the detector is linked to the size of the structural model that needs to be used. The set of part detectors is specified by the operator, including the number of stages to be used for training. Each part detector will produce a set of detection maps. Figure 3.4 shows an example of one sample creating a detection map for each part detector for one scale.

In this thesis the feature set was limited to 2-tuple part detections. To identify positive and negative examples of this feature set, positive and negative images are specified by an operator for the training algorithm to search. The specified set of part detectors is run over the positive and negative images. Positive images are resized to the specified size of the structural model before the part detectors are run, and the negative images use a windowing approach. The windowing approach used with the negative samples creates many sample from one sample image. The resultant detection maps are discretized and only scale 1.0 detections are searched for detection pairs.

Detection pairs are found within a set of maps for each sample. If three parts were being used to detector a whole object, parts A, B, and C, all pairs of detections for parts A, B, and C would be stored as possible features. Figure 3.6 shows two examples of features defined in the feature space of part observations. The red circles denote the two part observations that compose a feature.

Each feature has an associated weak classifier that will search a provided discrete image set for

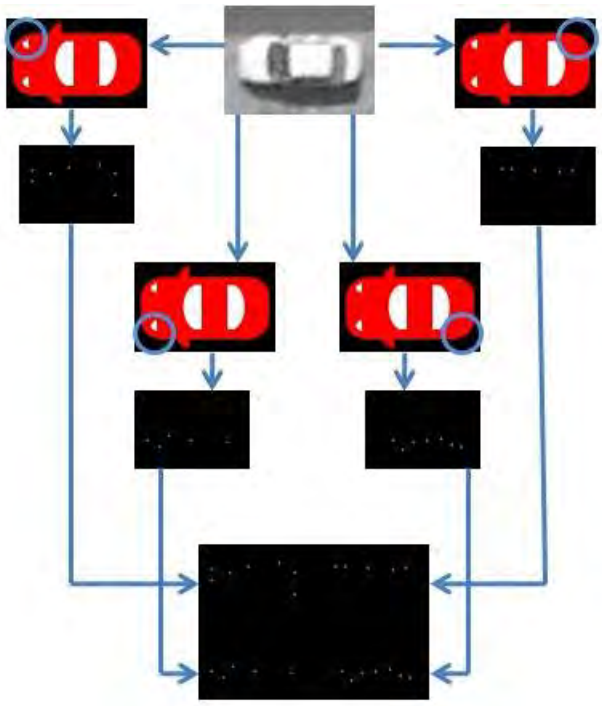

Figure 3.4: Graphical illustration how the four corner detectors are applied to an image.

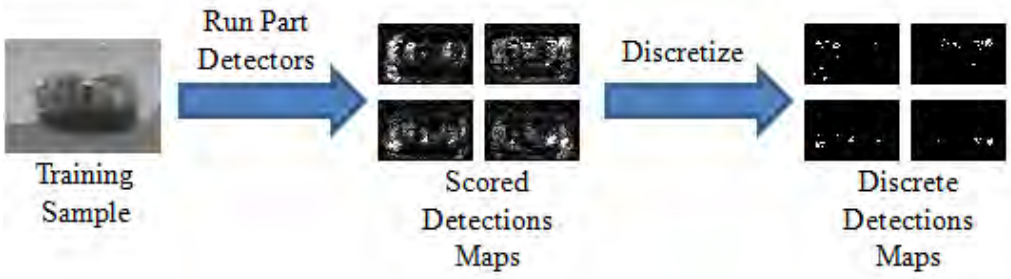

Figure 3.5: Illustration of how the discrete map sets are generated for a sample.

an image and determine if the feature exists. For this thesis all weak classifiers were implemented to return either 0 for not present, or 1 for present. A second iteration is made through all training samples to compute the prediction of all weak classifier returns for all samples. These predictions are used as input to an adaboost algorithm that generates a decision tree.

3.4.6 Performing Parts-base Detection

The trained decision tree contains the logic for the structural model, at the specified size of the model. To account for objects that appear over a range of scales, a set of scored detection maps

Figure 3.6: Two examples of features defined by a pair of part detections.

is returned for each part detector. A separate file containing all features found while training the structural model is used to compute all features for an image, at each scale increment returned in the sets of detection maps. A scale is searched if all parts have a detection map of the specified search scale, and the size of all parts' detection maps at the scale are equal to or greater than the size of the structural model. This means that the minimum and maximum scale factor searched for an object was defined by the most limiting scale factors defined in the set of part detectors. For detections maps with dimensions (width and/or height) greater than the dimensions of the structural model, a windowing approach is used where column and row offsets are increased until the entire detection map has been searched.

To adjust the recall and false positive rate, a threshold value can be set for the parts-based detector. This threshold value is the value at or above which the weak classifiers will accept an observation. For a feature to be present, both locations in the 2-tuple must have values equal to or greater than the threshold value specified.

For a given offset within a scale of detection maps, once all weak classifiers have been calculated, the calculated predictions are passed to the decision tree. The decision tree returns its prediction as a 0, if the vehicle is not present, or 1, if the vehicle is present.

CHAPTER 4:
Experimentation

The UAV used for these experiments was a customized Sig Rascal 110 ARF remote controlled aircraft, shown in Figure 4.1. The Unmanned Systems Lab at Naval Postgraduate School maintains and operates a small fleet of these aircraft that have been outfitted with Piccolo flight control systems and PC-104 boards to perform automated flight operations and payload operations. Wave relay is the network used by the Rascal UAV when operating at the Camp Roberts testing area [34], where the field experiments for this thesis were conducted.

Figure 4.1: Picture of the Sig Rascal 110 ARF remote controlled aircraft.

4.1 Vehicle Detection in Aerial Imagery

The initial vehicle detection experiment transmitting the imagery collected by the UAV to a ground station desktop for object detection. For this experiment one NPS Rascal was outfitted with two PC-104 boards and a gimbled camera to collect imagery. Both PC-104 boards were Advanced Digital Logic MSM800XEV, with a 500MHz AMD processor and 256 MB of soldered memory. The camera was a Cannon G6 PowerShot, capable of taking 12MP images. Customized open source software commanded the camera over USB during the flight. The gimble was controlled by the flight operation PC-104, and accounted for the UAV's parameters such that the camera was pointing directly down to allow for collection of still imagery from a near nadir aspect.

The ground processing station was a Windows XP desktop computer with an Intel Core 2 quad core CPU running at 2.40 GHz and 3 GB of physical RAM installed. During testing, a standard set of desktop processes continued to run, for example, anti-virus and default video card drivers.

4.1.1 Bandwidth Performance of Wave Relay

To test the available bandwidth of the wave relay network at Camp Roberts a fixed size file was continuously transmitted from the UAV to the ground station. During these repeated transmissions periodic monitoring of network performance at regular intervals provided data points to allow assessment of the bandwidth nominally available between the Rascal UAV and the controlling ground station.

4.2 Onboard Detection

Based on the processing times from the ground station processing experiment, the MSM800 PC-104 board would not provide sufficient computing power to perform vehicle detection onboard the UAV. The Advanced Digital Logic AGL945PC was selected to replace both MSM800 boards previously installed on the Rascal. The flight control PC-104 had used a version of Real DOS to provide real-time gimble control, and in the migration to the single board running linux the real-time gimble control was dropped due to time constraints to get the UAV flight ready.

4.3 Accounting for In-plane Object Rotation

The objective of this test was to determine the sensitivity of a Viola-Jones detector to in-plane rotations. Two methods of accounting for in-plane rotation of the vehicle (in the image plane) will be evaluated in this test. Metrics to be used in evaluation will be recall, false positives (per image), and average processing time for a given set of images. For a comparison of recall between the two detection methods, recall will be evaluated as the independent variable with false positive rate (per image) as the dependent variable.

4.3.1 Aligned Training Set

Using an aligned training set, as shown in Figure 4.2, a vehicle in a test image can be aligned with the detector by rotating the image. In these experiments the images being evaluated were rotated in five degree increments. For each rotational increment of the evaluation image, the detector will be applied. Figure 4.3 is an example of an image that has been rotated 145 degrees in preparation for applying the detection algorithm.

4.3.2 Rotated Training Set

This method will use the same negative image set but will increase the positive image set by cropping the images in five degree increments, from their source image. The reason for cropping

Figure 4.2: This shows the training set as it is provided to the OpenCV haar cascade training algorithm for the non-rotated cascade.

is to eliminate black areas from the positive image set that would be consistently present, and could be retained by the boosting algorithm as features to be used in detecting the objects.

4.4 Parts-based Detection

4.4.1 Evaluation of Selected Parts for Whole Object Detection

Using four discrete corner detectors — detectors that returned a 1 if a window of the image passed the threshold of all cascade stages and a 0 otherwise — detection maps were created that represented the positive detections for training and test images. 200 positive vehicle images and 160 negative images were used to train four types of data mining models: decision tree, support vector machine (SVM), bi-directional Kohonen self-organizing map, and multiple neural networks with varying units in the hidden layer. The trained models were then applied to the training set and to a test set of one hundred positive vehicle images and 72 negative images. For this portion of testing all vehicles were rotated such that the hood faced to the left in all images.

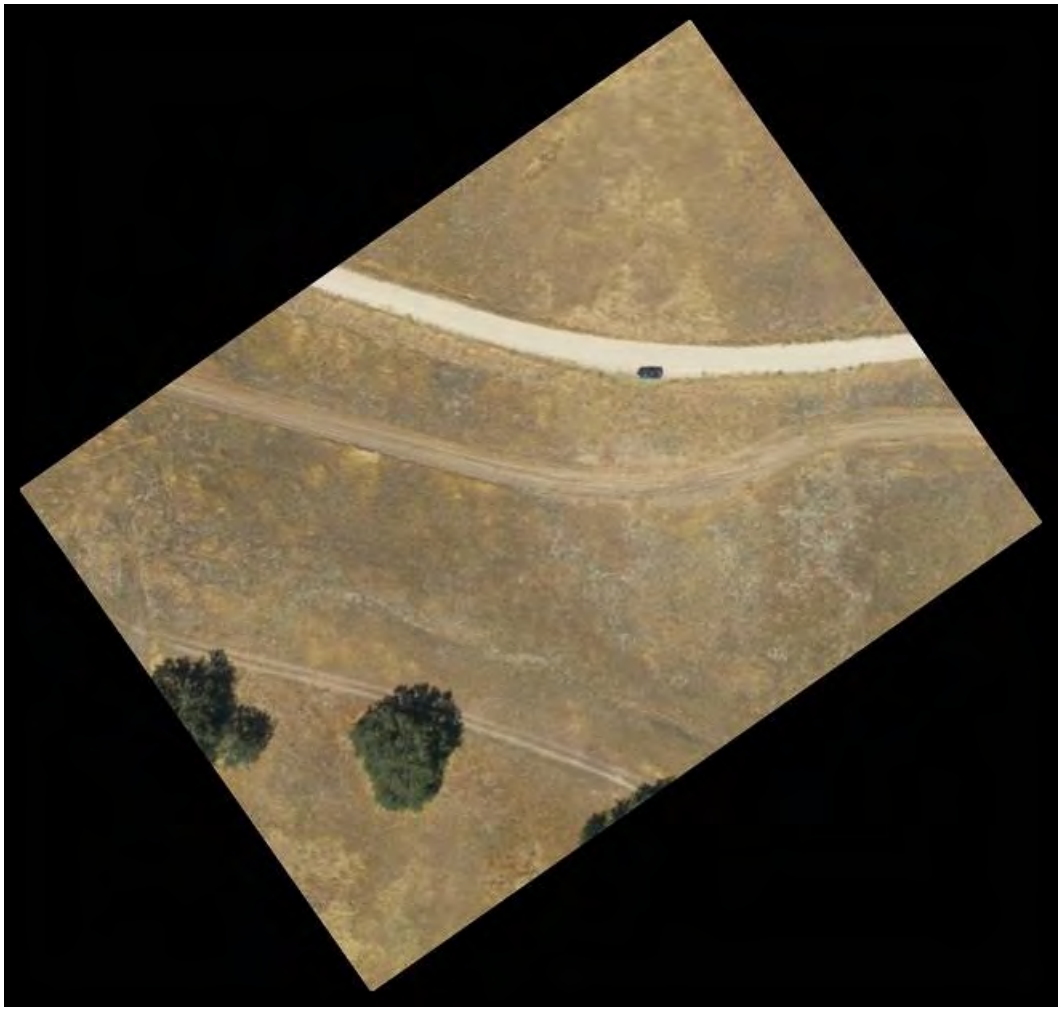

Figure 4.3: This shows the training set as it is provided to the OpenCV haar cascade training algorithm for the non-rotated cascade.

These results were returned in a matrix structure of equal to the standardized height minus detector height and width minus detector width. All training and test images used were resized to the same size, 73 pixels in height and 50 pixels in width, which is the native size of the whole vehicle. The returned matrix is reduced in height and width by the size of the detector because a positive return is indicated at the center of the detector. Had the returned matrix been equal to the size at which the image was searched it would have a border of zeros at the top and bottom equal to half the height of a corner detector and at the left and right sides of half the width of a corner detector. The corner detectors are 13 pixels in height by 13 pixels in width, so the resultant detection matrix is 60 pixels in height by 37 pixels in width. This results in 2220 pixels, which are also the possible locations for a detection in the image. Each image is searched

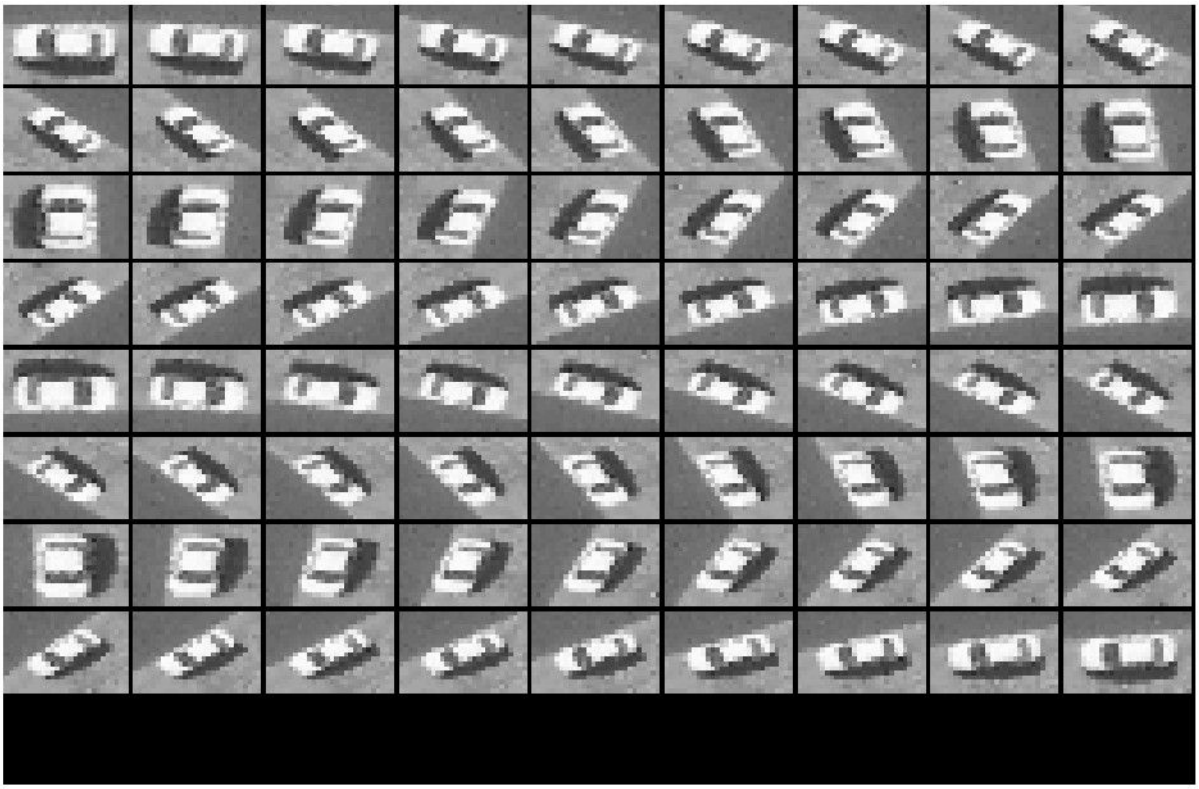

Figure 4.4: This shows the subset of the training set for the rotations applied to the first training image.

by all four corner detectors resulting in four matrices of 2220 pixels, or 8880 pixels of corner detection information.

Figure 3.4 shows how the four corner detectors, applied to a single image, produce four observation maps. Each observation map is the collection of observations in the image for one of the four corners. These maps are translated from a two dimensional matrix into a one dimensional array as input to the data mining models. The resultant measures of recall and false positive rate can be used to validate whether the four vehicle corners are sufficient information to predict the presence or absence of a vehicle.

4.4.2 Structural Model

A structural model was trained using 100 positive images, and windowing through five negative images, each 4000 by 3000 pixels. Testing was performed using the same negative image set as for the aligned and rotated cascades, but reduced by 33% in size due to the run-time complexity of testing the structural model, and the same 100 positive images. This detector does not use

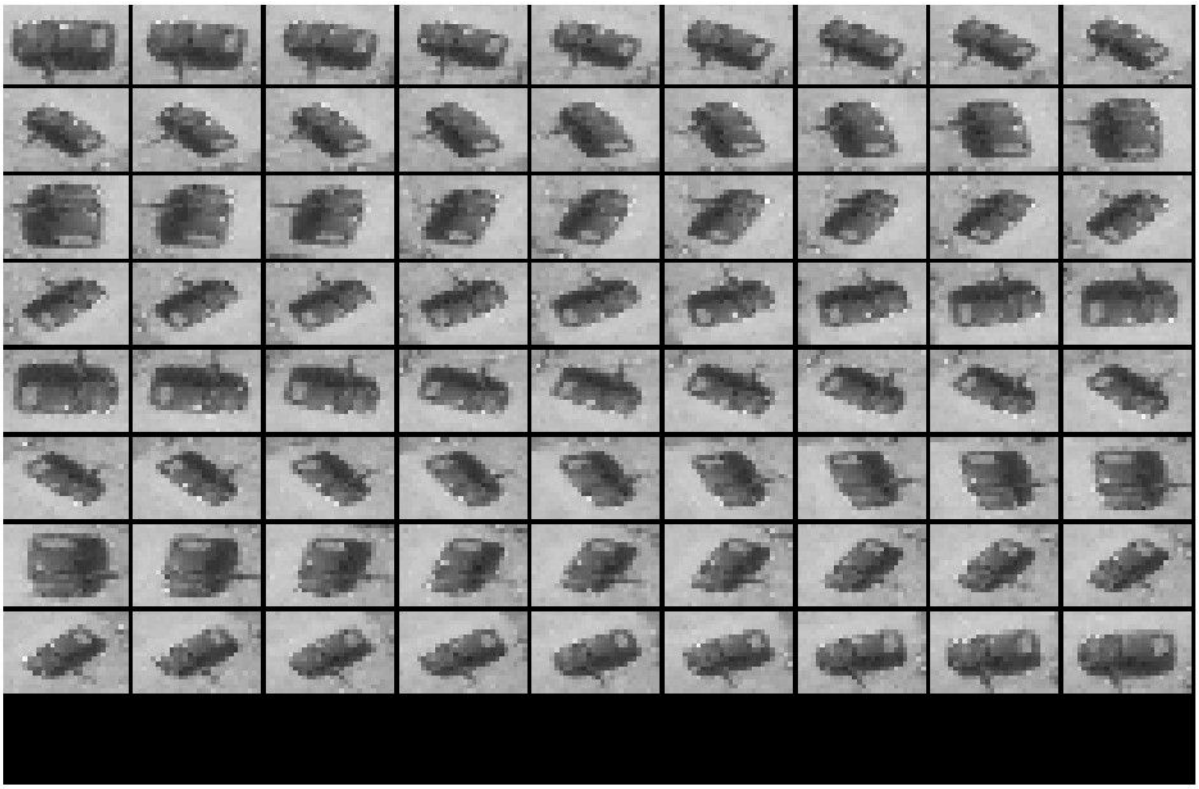

Figure 4.5: This shows the subset of the training set for the rotations applied to the second training image.

post-processing to combine object detections.

To produce a ROC curve, the threshold value of the parts-detector was varied between 0.0 and 1.0, real values, as the parts-based detector searches for vehicles. The structural model utilized scored detection maps. By varying the threshold of a part observation, the number of required stages a particular pixel location must complete is varied, proportional to the number of stages in the part detector.

CHAPTER 5:
Results

5.1 Vehicle Detection in Aerial Imagery

The goal of this experiment was to show that it is possible to recognize vehicles in aerial imagery using a Viola-Jones detector. For this experiment, the whole vehicle detector, trained with rotated positive samples, was used. Table 5.1 shows the results for four experimental flights. All vehicles present in an image were counted manually to represent the ground truth. Vehicles

Flight	Images Taken	Vehicles Present	True Positives	Missed Positives	False Positives	Recall	FPR (per image)
2010-08-08	714	715	433	282	321	0.6056	0.4496
2010-08-11a	760	95	82	13	69	0.8632	0.0966
2010-08-11b	719	19	16	3	59	0.8421	0.0826
2010-08-12	414	9	7	2	147	0.7778	0.2058
Overall	2607	838	538	300	596	0.7721	0.2086

Table 5.1: Detector detection performance during testing at Camp Roberts in conjunction with TNT 10-4.

present in more than one image were counted each time. A vehicle was counted if more than 50% of the vehicle appeared in an image. The count of vehicles appearing in an image was summed for all images taken during the flight and presented in the "Vehicles Present" column. For example, if the same car appeared in five images, that vehicle was counted five times. Each detection that contained a vehicle was manually verified as a true positive. A cropped detection containing multiple vehicles was counted as the number of vehicles it contained. For example, a cropped detection containing three vehicles was counted as three detections. Post-processing combines nearby detections, so any one cropped area may be a collection of multiple detections. "Missed Positives" are the number of vehicles present in imagery that were not detected, calculated by subtracting true positives from the vehicle detections for the flight. Each cropped detection not containing a vehicle is counted as a false positive, and all such cropped detections were summed for the false positives for a given flight. Recall R is calculated as the ratio of true positives P_T to vehicles present V, $R = \frac{P_T}{V}$. False positive rate (FPR) F is the ratio of false positive detections P_F per image I processed, $F = \frac{P_F}{I}$.

The average recall was 77.21% with an average false positive rate at 0.21 false positives per

image. Flights 2010-08-11a and 2010-08-11b had significantly lower false positive rates and higher recalls compared to the other two flights.

5.1.1 Speed Performance of Ground Station Processing

Table 5.2 shows the image access and processing statistics for the four flight experiments. ”Raw

Flight	Raw Total (kB)	Crop Total (kB)	Mean Access (ms)	Std Dev Access (ms)	Mean Processing (ms)	Std Dev Processing (ms)
2010-08-08	3914757	18656	504.40	46.17	1494.11	149.55
2010-08-11a	4242232	4198.6	542.70	122.92	1532.57	156.56
2010-08-11b	4165580	2394.8	500.65	41.36	1473.77	155.13
2010-08-12	2532752	4589.6	511.79	46.04	1449.56	205.83
Overall	14855321	29839	514.89	64.12	1487.50	166.77

Table 5.2: Detector speed performance during testing at Camp Roberts in conjunction with TNT 10-4.

Total” is the total storage space of all images taken on the flight in kilobytes on disk. ”Crop Total” sums all cropped detections for the flight in kilobytes. ”Mean Access” and ”Std Dev Access” are the mean time required to access the files in the flight in milliseconds and the standard deviation of the time required to access the files in the flight in milliseconds, respectively. ”Mean Processing” and ”Std Dev Processing” are the mean time require to perform the detection algorithms on the images in the flight in milliseconds, and the standard deviation of the time required to perform the algorithms on the images in the flight in milliseconds, respectively.

On average it took 514.89 ms to read an image from disk and 1487.50 ms to run the vehicle detection algorithms on the image. Flight 2010-08-11a had a larger mean access time compare dot the other two flights, but flight 2010-08-11a’s standard deviation for access time was about three times that any of the other flights.

5.1.2 Bandwidth Performance of Wave Relay

Figure 5.1 shows the observed network bandwidth available to the UAV through an experimental flight. This throughput graph is comprised of a collection of data points collected at a regular interval during a specified flight period. During this testing above ground altitude was changed, but this information was not collected such that it could be associated with specific data points. The selected flight period did not involve take-off or landing evolutions. The x-axis is evenly spaced data collection points with no units, and the y-axis is the throughput for a given datum

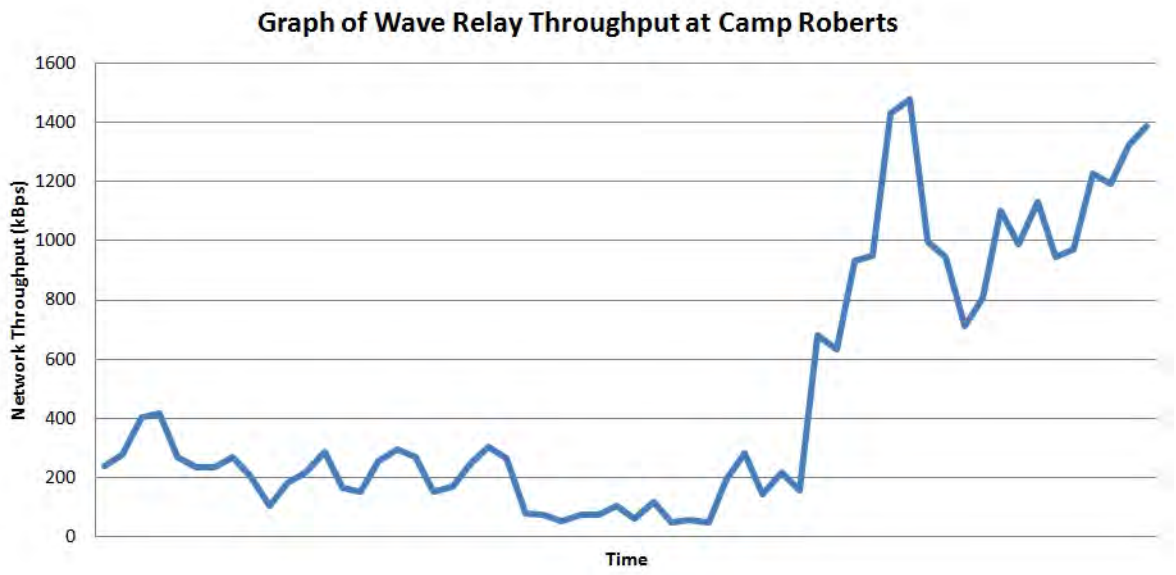

Figure 5.1: Graph of available network bandwidth over wave relay at Camp Roberts.

in kilobytes per second (kBps).

The graph in Figure 5.1 shows that the available bandwidth between the Rascal UAV and the ground controlling station over wave relay fluctuates over time. The bandwidth peak is caused by a reduction in altitude, which shows that the bandwidth is dependant upon altitude.

5.2 On-board Detection

During on-board testing the same detector with the same settings was used as in the previous vehicle detection experiment. Table 5.3 shows the detection results. The average recall of the two flights is 47.02%, but there is a large difference between flight 2011-05-06's recall of 73.54% and flight 2011-05-07's recall of 20.51%. The false positive rate if almost four times larger for flight 2011-05-07 than 2011-05-06 at 0.80 and 0.21 false positives per images, respectively.

5.2.1 Speed Performance of Onboard Processing

Table 5.4 shows the speed performance for conducting vehicle detection onboard the UAV. The mean processing time is 3057.16 ms, with no large discrepancy between the mean processing time of the two flights. The mean access time is is 607.87 ms, but flight 2011-05-07 has a much larger standard deviation for access time.

Flight	Images Taken	Vehicles Present	True Positives	Missed Positives	False Positives	Recall	FPR (per image)
2011-05-06	113	461	339	122	291	0.7354	0.2051
2011-05-07	398	1414	290	1124	570	0.2051	0.7983
Overall	511	1875	629	1246	861	0.4702	0.5017

Table 5.3: Detector performance during testing at Camp Roberts in conjunction with TNT 11-3.

Flight	Raw Total (kB)	Crop Total (kB)	Mean Access (ms)	Std Dev Access (ms)	Mean Processing (ms)	Std Dev Processing (ms)
2011-05-06	472171	17167.5	548.73	73.98	3020.46	353.45
2011-05-07	2015450	20700.1	667.02	551.67	3093.87	395.62
Overall	2487621	37867.6	607.87	312.82	3057.16	374.54

Table 5.4: Detector speed performance during testing at Camp Roberts in conjunction with TNT 11-3.

5.3 Accounting for In-plane Object Rotation

Figure 5.2 shows 72 receiver operating characteristic (ROC) curves. Each curve plots the false positive rate (per image) versus recall. The most desirable response curve depends on the exact implementation of the detector, but in general a response curve that has points with a higher recall and a lower false positive rate is preferred. Each curve represents the results of the detector using a varying number of cascade stages against a rotated data set. The 72 data sets are rotated in five degree increments, as denoted in the legend.

Figure 5.2 shows the large amount of variance that is observed as the same test data set is rotated. This shows that a detector trained with all positive images aligned in the same direction will be able to provide detection capable across a narrow range of variance in the in-plane rotation, with a significant drop off in performance once an object is rotated more than 10 degrees.

5.3.1 Rotated Training Set

The same test images were processed by the detector trained on aligned training images and rotated training images. 72 series of test images were used to test the 72 5 degree increments of the in-plane rotational space. Each series contained 100 test images rotated to the same orientation, and each series used the same source images from which to rotate and crop the 100 test images.

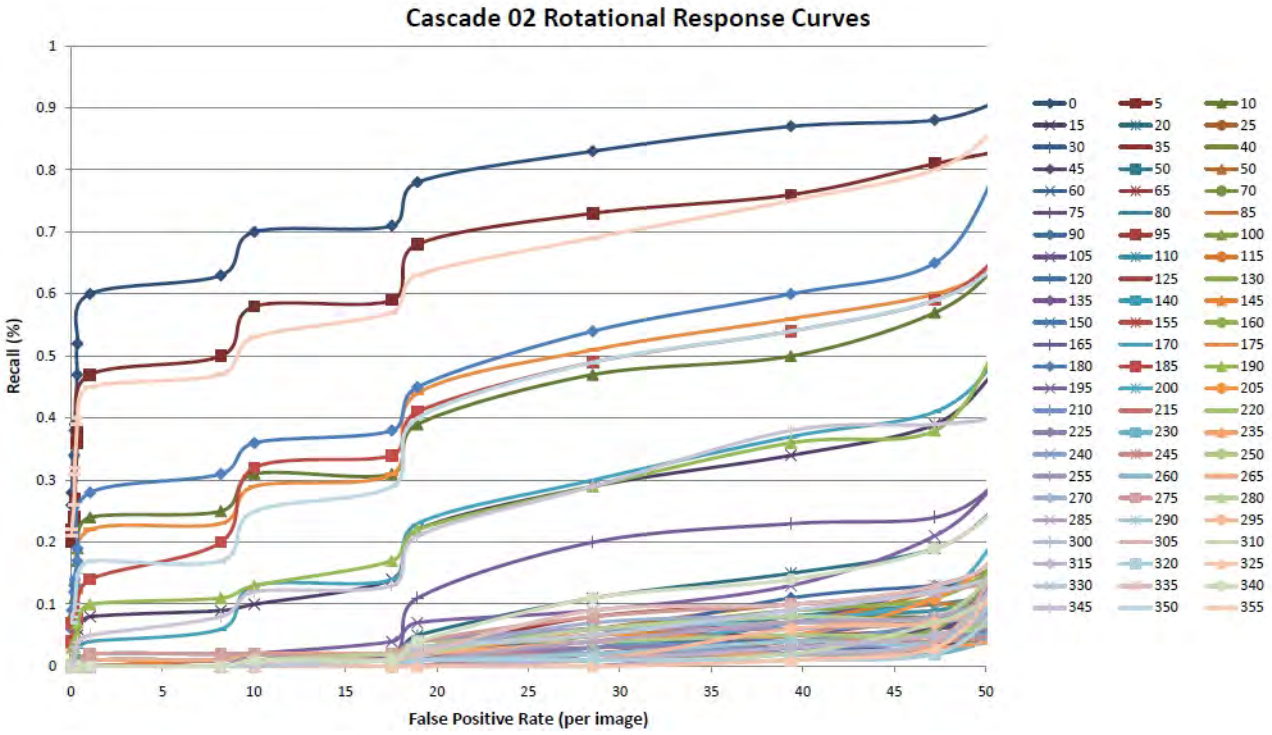

Figure 5.2: The graph above shows the performance of the detector trained with an aligned positive imagery set.

5.4 Parts-based Detection

5.4.1 Evaluation of Selected Parts

The following results show that observations of the four vehicle corners contain sufficient information for that feature space to be used to create a structural model that uses vehicle corner observations. Results in this chapter are for runs on the test data set, validation runs on the data set used to train the model are available in the Appendix A.

A two class confusion matrix is used to summarize the findings of all the applied data mining models. The predictions are the classifications as determined by the data mining model being summarized by the Table. Truth is the actual class to which the test sample belongs, and is a hand annotated label.

Decision Tree

The trained decision tree in Figure 5.4 shows the splits and values applied to the test data set, resulting in the confusion matrix shown in Table 5.5. Recall for the decision tree model was

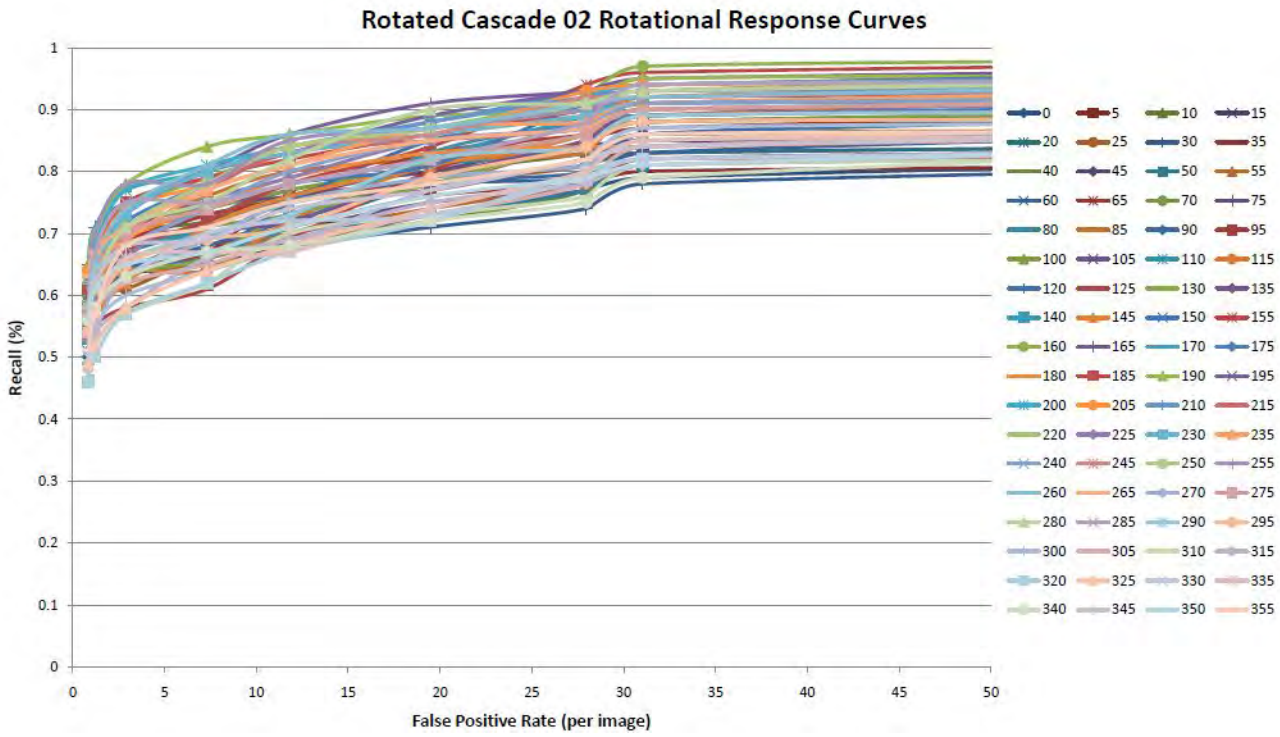

Figure 5.3: The graph above shows the performance of the detector trained with a rotated positive imagery set.

37%, false positive rate was 1.39%, and precision was 97.37%.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	71	63
	Vehicle Present	1	37

Table 5.5: Confusion matrix showing the results of the decision tree model.

Support Vector Machine (SVM)

The SVM model achieved a recall of 77%, false positive rate of 1.39%, and precision of 98.72%.

Kohonen Self Organizing Maps

Figure 5.5 is a graphical representation of the bi-directional Kohonen self-organizing map applied to the training data. In this case the self-organizing map has created 48 clusters, of which three hold the negative samples, labelled 0, and the remaining hold the positive samples, labelled

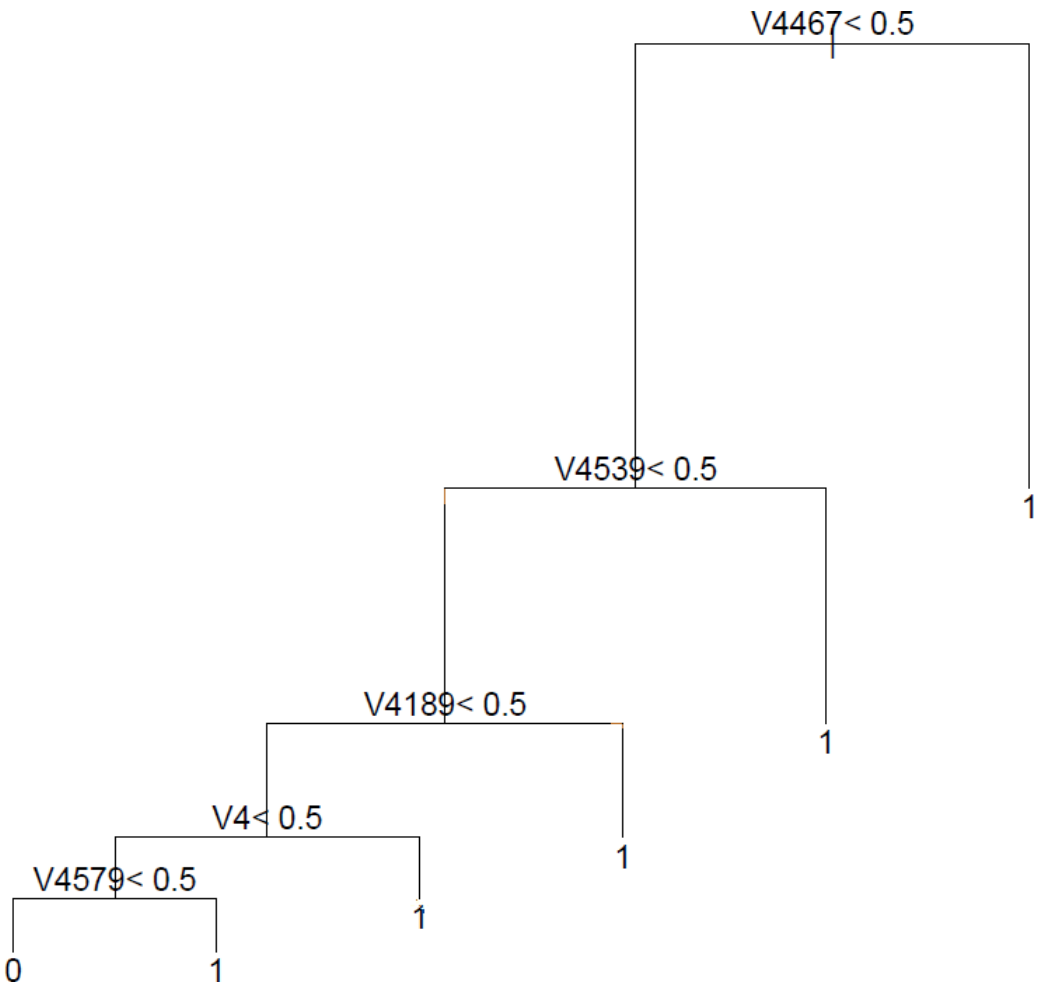

Figure 5.4: Graphical illustration of the trained decision tree model.

1. Of the three clusters containing negative samples, one contains only negative samples, and the other two have some confusion, they contain both positive and negative samples. This representation illustrates the results shown in Table A.3. The results summarized in Table 5.7 are organized into the same clusters. Applying the trained bi-directional Kohonen self-organizing map model to the test data set results in a recall of 79%, false positive rate of 1.39%, and precision of 98.75%. The confusion matrix if shown in Table 5.7.

Neural Network

The confusion matrix in Table 5.8 was collected by the neural network trained with three units per hidden layer when applied to the test data set. Recall was 90%, false positive rate was 1.39%, and precision was 98.9%.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	71	23
	Vehicle Present	1	77

Table 5.6: Confusion matrix showing the results of the SVM model.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	71	31
	Vehicle Present	1	69

Table 5.7: Confusion matrix showing the results of the Bi-Directional Kohonen Self-Organizing Map model.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	71	10
	Vehicle Present	1	90

Table 5.8: Confusion matrix showing the results of the Neural Network with 3 units in the hidden layer.

5.4.2 Structural Model

Figure 5.6 compares the parts-based detector performance with the aligned Cascade 02 detector's performance. The ROC curve for the parts-based detector, using the developed structural model, is graphed on a linear scale in appendix A, Figure A.1. The false positive rate is expressed in the percent of false positive detections returned (P_f) to the number of windows searched (N_w), $\frac{P_f}{N_w}$. Recall is a percentage of the true detections divided by the total number of vehicles present in the data set.

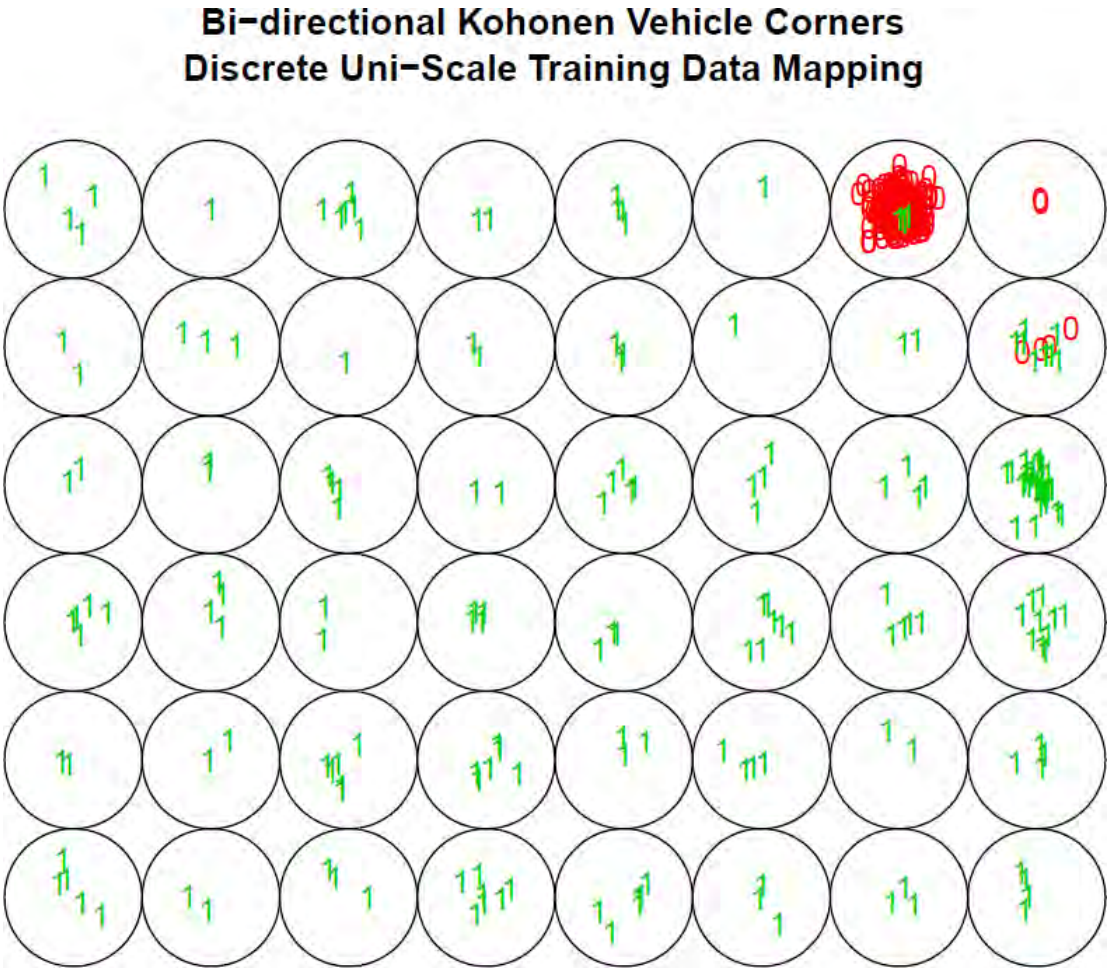

Figure 5.5: Graphical illustration of the trained Bi-Directional Kohonen Self-Organizing Map model.

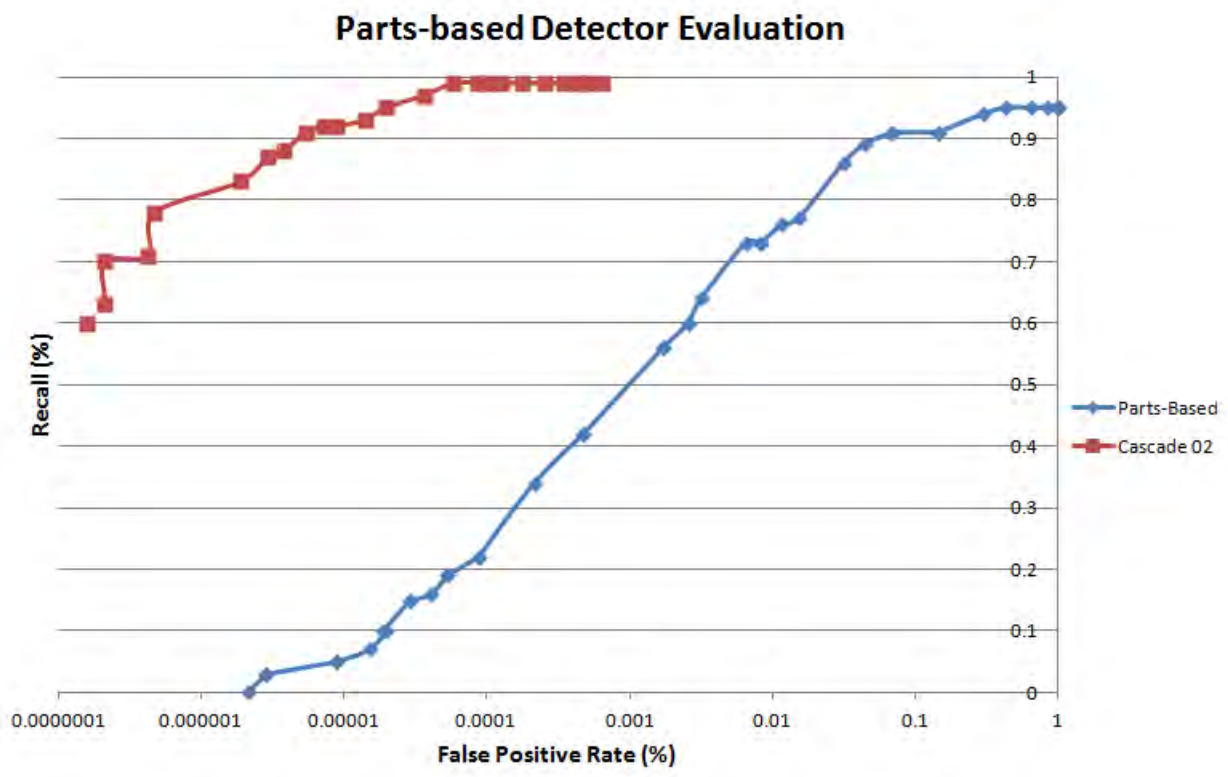

Figure 5.6: The ROC curve for the parts-based detector compared to the aligned Cascade 02 detector, on a logarithmic scale.

CHAPTER 6:
Discussion

6.1 Vehicle Detection in Aerial Imagery

Table 5.1 shows that vehicle detection is possible in aerial imagery. The recall varied across the four fights due to variations from image to image. Although the time of day, above ground level altitude, and detector settings were constant across all pictures, the UAV itself does not always exhibit consistent flight and camera platform parameters. For example, the gimble may be in motion while a picture is being taken, which could result in both blur and a viewing aspect that is not consistent with the training samples provided to the detector mostly nadir. Motion blur and changes in viewing aspect mean that the specific features for which the boosted cascade was trained may not be present, and therefore the detector will not be able to identify the presence of a vehicle in the image.

The experiments in this thesis used a limited scale factor for the sliding window detector scan across the image to reduce the false positive rate and processing times for each image. While this was a manual input to a configuration file, read only when the detection program was initialized, it is possible that this parameter could be automated. To automate limiting the search scale range, a range of vehicle sizes would need to be specified by the operator(s) and the automated system would require the pixel resolution to calculate the appropriate minimum and maximum search scale factors. With those two pieces of information the range of size in pixels can be calculated, which can be used against the trained detector size to determine the scale range over which the detector should search.

6.1.1 Speed Performance of Ground Station Processing

An image was collected once every three seconds, which is greater than the mean processing times shown in table 5.2. The detection algorithm can keep up with the limits of the platform, but slow network performance meant the detector was often waiting for the next image to be downloaded.

6.1.2 Bandwidth Performance of Wave Relay

Changing altitudes did have a noticable effect on the network bandwidth available on wave relay. This can help to explain some of the frustratingly slow transmission of the large image

files, which resulted in a continuously growing backlog of imagery waiting to be transmitted. The desktop computer responsible for processing the imagery was often waiting while the UAV continued to collect more imagery.

6.2 Onboard Processing

Although the same detector was used for the onboard detection experiments and the ground based processing experiments, the descriptive statistics of the two flights were very different. Flight 2010-05-07 had a much lower recall, 20.51%, compared to flight 2010-05-06, 73.54%, or any flight conducted during the TNT 10-4 experiments, an average of 77.21% as shown in table 5.1. Reviewing the imagery, there appears to be some halo and blurring around the edges of the vehicles that was not present to the same extent on the other 5 flights. Object boundaries and edges contain a lot of information, and many potential features for a haar-like feature based detector to utilize. This, combined with additional blurring of the imagery, is most likely the cause of the reduced recall.

6.2.1 Speed Performance of Onboard Processing

Comparing table 5.2 to Table 5.4 the on-board detection did take approximately twice as long to perform as detection on a desktop computer at a ground station. Differences in access times was negligible between the two detection pipelines.

6.3 Accounting for In-Plane Object Rotation

For optimal performance rotation increments of 5 degrees should be used, but from figure 5.2 rotational increments of more than 10 degrees will most likely result in significantly reduced recall.

6.3.1 Rotated Training Set

A detector based on a rotated training set will contain more features, and therefore require more CPU cycles to process an image, however Figure 5.3 shows that the variance in recall across the range of possible in-plane rotations is reduced such that only one pass of the image is required. The recalls of the two detectors, at a selected false positive rate, should be used to directly compare Figures 5.2 and 5.3.

Training a detector with the additional samples (14400 positive samples in 72 orientations versus the 200 positive samples used for the aligned detector) results in a modest increase of

training time off-line. However, the detector trained with rotated positive samples will produce results comparable to a detector trained with aligned positive sample without the need for image rotation. Image rotation is computationally expensive process at run time, can introduce artifacts caused by the interpolation method, and increases the pixel count by adding the black areas seen in Figure 4.3.

6.4 Parts-based Detection

6.4.1 Evaluation of Part Selection

The decision tree model performed poorly, as shown in table 5.5. Its poor performance is most likely due to the fact that the tree used only pixel locations that did not contain positive detections in the trained sample. This becomes more clear when comparing the results of the validation set in table A.1 and the test set in table 5.5, which shows that the model consistently achieves a lower false positive rate. The decision tree model performed significantly better on the validation set than the test set, which is an indication that the decision tree model is over trained. The results of this model on the test set would seem to indicate that observed vehicle corners do not contain enough information to correctly determine if a vehicle is present or absent from a given sample image.

The SVM, bi-directional Kohonen self-organizing map, and neural network all showed improved performance of the decision tree model, with the later model(s) improving over the previous model(s). The neural network had the best recall performance on the test set, although its training set recall was only the third best. This is a good indication of a model that is more robust to variance in the positive object class. All three sets of results from these models support the hypothesis that vehicle corner observations do contain sufficient information to perform object detection.

6.4.2 Structural Model

The ROC curve shows that the parts-based detector does detect vehicles. Due to the high false positive rates, this parts-based detector is not usable without some post-processing technique. This detector does reduce the searchable area of a raw image, and would be useful for input into a larger detector that requires more computing power but with more discriminating power to filter out the false negatives. The part detector are small and would be simpler to develop in a hardware solution. Likewise, the features used to describe the structural model are simple and

could also be implemented into hardware. This would significantly speed up the detection process using this technique, and perform the majority of the detection algorithm prior to requiring a more general purpose processing unit.

CHAPTER 7:
Conclusions

7.1 Detection of Vehicles in Aerial Imagery

12MP images (3000 x 4000 pixels) can be processed on a desktop PC (Intel quad core, 2.40GHz) at speeds exceeding the capture speed of one image every three seconds. The limiting factor for analyzing such large images is the network bandwidth rather than the processing speed, because the image must complete downloading to the desktop before processing can begin.

In the ground station detection and onboard detection experiments the ability to detect the vehicle in the image means that the GPS coordinates of the detection can be determined by interpolation. This specific experiment interpolated the GPS coordinates of a detection at the center of the detection from the GPS coordinates available at the corner of each image. Flight control software on the non-payload PC-104 calculated the GPS coordinates of the image corners, and inserted this information into the xif data jpeg for the detection software to use as a post-processing step.

As another portion of the experiment a human operator confirmed the detections before passing them via a UDP message to another computer program that consumed the detection location information to form a predictive movement model. From this predictive model it was then possible to recommend a flight plan that when confirmed by a human operator, directed the UAV where to fly and when to collect imagery. The importance of this experiment is the semi-autonomous nature of developing a model and recognizing a flight plan that could regain the contact, instead of the human operator recognizing and flying a flight plan.

7.2 Onboard Object Detection

The same algorithm used to process images on the ground can be implemented on a UAV. The computing hardware requirements for the UAV increase due to perform the analysis on-board, and as hardware is upgraded to obtain the necessary processing power the electrical power requirements increase. As processor architectures continue to shrink, the number of available cores and processing speeds will increase such that the UAV to desktop speed gap will shrink.

The onboard detection experiment showed that it is possible to significantly reduce network bandwidth by at least one order of magnitude by sending cropped detections instead of the original raw image. Reducing network usage means that this resource is available for other assets operating in the area.

7.3 Accounting for In-plane Rotation

Rotating the positive training samples in a consistent method can create a Viola-Jones detector, using haar-like features, that is robust to rotational variance. While the offline training time of the detector increases, the run-time advantage is significant and conserves computational resources. This method of training to address the weakness of haar-like features in handling rotation is a key component of the algorithm that enables on-board detection. The additional pixels produced at each rotation, the processing time to perform the rotations, and the several passes required by each detector would require additional power not available on-board a UAV, compared to the rotationally trained detector.

While this thesis has shown a rotated training set of positive images to work for a whole image detector, it remains to be proven that this technique will work for part detectors. Part detectors are a critical component of a parts-based detection method, because the individual part detectors provide the observations to the algorithm for deciding if an object, or possibly which object, exists in a test image.

It is also possible that the number of features in each cascade stage can be reduced for some objects if the full 0 to 360 degree range of rotation must be provided. For example, rotationally symmetrical objects may only require 180 degrees of rotated positive imagery, and therefore fewer possible features, resulting in a smaller, faster cascaded detector.

7.4 Parts-based Detection

The parts-based detection results show that it is possible to train a and apply a structural model to a mapped part observations, and use the model for object detection. The tests in this thesis were only on vehicles, so testing is still necessary for articulated objects. All vehicles were normalized to face in one direction, so the part detectors and structural model needs to be advanced to account for in-plane rotation. However, this thesis did show the ability to detect across scale, so the method of structural model development introduced in this thesis is robust to scale variance.

The specific detector employed in this thesis used a binary tree, so all discovered features were used in by the decision tree that adaboost created. Using the adaboost to develop a boosted cascade, where each stage is bound to use features that achieve performance within a specific range for false positive and recall metrics, would reduce the number of features to be calculated and compared. All features were also calculated for the entire tree before utilizing the decision tree, so calculating features as needed would reduce memory requirements, and eliminate the calculation of features that are not used.

The parts-based detector presented in this thesis has two modules that can be resized the part detectors and the window size of the structural model. Knowing the minimum size of an object to search for can aid in making the correct decision for these sizes, but it may be necessary to have two differently sized parts-based detectors, each of which is optimized for a range of object scales. It may be possible to determine optimal detector and window sizes by first using the data mining methods used to validate corners as reasonable parts on which to base a parts-based vehicle detector. Varying the size can vary the model, and comparing results with the bi-directional Kohonen self-organizing map and neural network models, a quick estimate could be obtained for the recall and false positive limit of a parts-based detector operating with a particular combination of detector and window sizes.

7.5 Operational Implementation

This thesis proved that detections can be performed real-time, however, the same algorithms used in this thesis can be implemented for forensic purposes. The forensic application has the ability provide back-up if an intelligence, surveillance, and reconnaissance (ISR) mission is required to be flown and no UAV is available with the computation power to perform onboard detection. Although a dedicated operator may be watching a streaming video or reviewing still imagery, that operator may miss crucial data, and a forensic implementation can provide an unbiased second opinion. Detection over a known imagery set can also be useful for testing a new detector or algorithm against a known detector.

The increased computing hardware necessary for on-board detection is added to a UAV, the size, power requirements, and price increase. For one or more of the previous three reasons, it is likely that not all UAVs will be able to perform onboard detection. As a result, UAVs should be matched to the mission so resources are not wasted. It is very possible that for a particular

mission full motion video is needed, and a dedicated analyst will be reviewing the video as it is streamed. This mission is most likely best flown by remote, which would mean that the onboard data retention is not advantageous in this case. A scenario like this can occur, but should be the exception because it is manpower intensive and does not scale well for multiple UAV missions.

Onboard detection, the resultant data, and the use of this data by autonomous flight functions could reduce the workload of a UAV operator to the point where one operator can safely operate several UAVs without the operator becoming tasked. This scenario is scalable, and uses much less manpower. Additionally, the manpower that is currently used to monitor UAV collection can be retasked from the monotonous flight and continuous monitoring of instruments. This manpower becomes available for providing forceful backup to ensure the goals of the mission are met, not just safe flight of the UAV. Supervisors of the operator are freed from providing immediate backup to function in a tactical and operational status, and can concern themselves less with the operation of the UAV and instead focus on the productive employment of the UAV as a reconnaissance asset. This will increase potential operational gains without increasing operational risk.

The parts-based solution can potentially make use of parts to define articulated objects, by using the weighting of features to allow for multiple configurations. It is also possible that by including general class and specific sub-class parts, the parts-based technique can use the information in the intermediate feature set to identify a class and sub-class object. An example is a detector that identifies the presence of a vehicle, but then additionally identifies the type of vehicle detected (i.e. a vehicle that is a car, or a vehicle that is a pick-up truck). The advantage of this detector is only one detector is needed, and the sub-class decision is made in one process, instead of using multiple detectors and a post-processor to determine the best quality detection of the detections identified by the multiple detectors.

This thesis demonstrates algorithms that have the ability to utilize information collected onboard the UAV in a way that can allow the UAV to become semi- or fully autonomous instead of remotely piloted. For example, the UAV could be instructed to follow the first vehicle that it encounters and collect imagery until the vehicle stops, then return to base. Onboard processing of imagery shifts the problem from situational awareness of the UAV to sufficient operational planning about what the goals of a flight are and how completion is measured. If a UAV is

unable to operate autonomously, then a significant tactical weakness is our ability to communicate. The ability of an enemy to deny clear communications with a friendly UAV means the UAV technology will have been defeated. To utilize a UAV in a communications denied area, it must operate autonomously.

THIS PAGE INTENTIONALLY LEFT BLANK

REFERENCES

[1] G. Dorkó and C. Schmid. Selection of scale-invariant parts for object class recognition. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceesings ICCV'03*, pp. 634–639. IEEE, 2008. ISBN 0769519504.

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[3] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1. Citeseer, 2001.

[4] J. Shi and C. Tomasi. Good features to track. In *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94.*, pp. 593–600. IEEE, 1993. ISBN 0818658258.

[5] K. Kaaniche, B. Champion, C. Pegard, and P. Vasseur. A vision algorithm for dynamic detection of moving vehicles with a UAV. In *2005 IEEE International Conference on Robotics and Automation, 2005. Proceedings of the ICRA 2005.*, pp. 1878–1883. IEEE, 2006. ISBN 078038914X.

[6] A. Miller, P. Babenko, M. Hu, and M. Shah. Person tracking in UAV video. *Multimodal Technologies for Perception of Humans*, pp. 215–220, 2009.

[7] K. Appiah, H. Meng, A. Hunter, and P. Dickinson. Binary histogram based split/merge object detection using FPGAs. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pp. 45–52. IEEE, 2010.

[8] C. Benedek, T. Szirányi, Z. Kato, and J. Zerubia. Detection of object motion regions in aerial image pairs with a multilayer Markovian model. *IEEE Transactions on Image Processing*, 18 (10):2303–2315, 2009. ISSN 1057-7149.

[9] S. Hinz. Combining local and global features for vehicle detection in high resolution aerial images. *Munchen, Germany, PF-2003-02*, 2003.

[10] S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 (11):1475–1490, 2004. ISSN 0162-8828.

[11] Z. Du, C. Guodong, L. Sun, J. Ji, and M. Xie. Local shape patch based object detection. In *2010 IEEE International Conference on Information and Automation (ICIA)*, pp. 1723–1728. IEEE, 2010.

[12] E.J. Bernstein and Y. Amit. Part-based statistical models for object classification and detection. 2005. ISSN 1063-6919.

[13] J. Sullivan, O. Danielsson, and S. Carlsson. Exploiting Part-Based Models and Edge Boundaries for Object Detection. In *Computing: Techniques and Applications, 2008. DICTA'08. Digital Image*, pp. 199–206. IEEE, 2008.

[14] X. Xia, W. Yang, H. Li, and S. Zhang. Part-Based Object Detection Using Cascades of Boosted Classifiers. *Computer Vision–ACCV 2009*, pp. 556–565, 2010.

[15] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2009. ISSN 0162-8828.

[16] X. Mao, F. Qi, and W. Zhu. Multiple-part based Pedestrian Detection using Interfering Object Detection. In *Third International Conference on Natural Computation, 2007*, volume 2, pp. 165–169. IEEE, 2007.

[17] J. Sivic, B.C. Russell, A.A. Efros, A. Zisserman, and W.T. Freeman. Discovering objects and their localization in images. 2005. ISSN 1550-5499.

[18] S.C. Wang and Y.C.F. Wang. Simultaneous Object Recognition and Localization in Image Collections. In *2010 Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 497–504. IEEE, 2010.

[19] T.B. Suja and M. John. Fusion based object detection. In *2010 National Conference on Communications (NCC)*, pp. 1–3. IEEE, 2010.

[20] A.Y.S. Chia, S. Rahardja, D. Rajan, and M.K.H. Leung. Structural descriptors for category level object detection. *Multimedia, IEEE Transactions on*, 11(8):1407–1421, 2009. ISSN 1520-9210.

[21] J. Li and N.M. Allinson. A comprehensive review of current local features for computer vision. *Neurocomputing*, 71(10-12):1771–1787, 2008. ISSN 0925-2312.

[22] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. *Computer Vision-ECCV 2004*, pp. 69–82, 2004.

[23] I. Kokkinos, P. Maragos, and A. Yuille. Bottom-up & top-down object detection using primal sketch features and graphical models. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 1893–1900. IEEE, 2006. ISBN 0769525970. ISSN 1063-6919.

[24] I. Fasel, B. Fortenberry, and J. Movellan. A generative framework for real time object detection and classification. *Computer Vision and Image Understanding*, 98(1):182–210, 2005. ISSN 1077-3142.

[25] S. Rao, NC Pramod, and C.K. Paturu. People detection in image and video data. In *Proceeding of the 1st ACM workshop on Vision Networks for Behavior Analysis*, pp. 85–92. ACM, 2008.

[26] N. Alt, C. Claus, and W. Stechele. Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 176–181. ACM, 2008.

[27] G. Passino, I. Patras, and E. Izquierdo. On the role of structure in part-based object detection. In *15th IEEE International Conference on Image Processing, 2008.*, pp. 65–68. IEEE, 2008. ISSN 1522-4880.

[28] Z. Ying and Q. Guang-jie. Car detection using codebook and Directed Graphical Model. In *2010 International Conference on Computer Application and System Modeling (ICCASM)*, volume 15, pp. V15–530. IEEE, 2010.

[29] A. Kapoor and J. Winn. Located hidden random fields: Learning discriminative parts for object detection. *Computer Vision–ECCV 2006*, pp. 302–315, 2006.

[30] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. 2005. ISSN 1063-6919.

[31] D. Park, D. Ramanan, and C. Fowlkes. Multiresolution models for object detection. *Computer Vision–ECCV 2010*, pp. 241–254, 2010.

[32] C.J. Hall, D. Morgan, A. Jensen, H. Chao, C. Coopmans, M. Humpherys, and Y.Q. Chen. Team OSAM-UAVS Design For The 2008 AUVSI Student UAS Competition. ASME, 2009.

[33] J. Jones and S.L. Garfinkel. Parts-based detection of ak-47s for forensic video analysis. 2010.

[34] M.R. Clement, E. Bourakov, K.D. Jones, and V. Dobrokhodov. Exploring network-centric information architectures for unmanned systems control and data dissemination. *AIAA Infotech Proceedings*, 2009.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A:
Appendix A

This appendix contains the additional confusion matrices for the data mining models developed to confirm the selection of vehicle corners as the component parts of a parts-based vehicle detector, and ROC curves.

A.1 Decision Tree

Validating the decision tree with the training set resulted in a recall of 87.5%, false positive rate of 0%, and precision of 100%.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	160	25
	Vehicle Present	0	175

Table A.1: Confusion matrix showing the results of the decision tree model applied to the training data.

A.2 SVM

The SVM model's recall was 99.5%, false positive rate was 77%, and precision was 98.72% when run on the training set.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	160	1
	Vehicle Present	0	199

Table A.2: Confusion matrix showing the results of the SVM model applied to the training data.

A.3 Bi-directional Kohonen Self-Organizing Map

Validating the training set with the bi-directional Kohonen self-organizing map produced a recall of 93.5%, false positive rate of 0%, and precision of 100%.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	160	13
	Vehicle Present	0	187

Table A.3: Confusion matrix showing the results of the Bi-Directional Kohonen Self-Organizing Map model applied to the training data.

A.4 Neural Network

The neural network was trained and tested with two and three units in the hidden layer. Validating the two unit hidden layer model with the training set produced a recall of 97.0%, false positive rate of 0%, and precision of 100%. Running the two unit neural network over the

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	160	6
	Vehicle Present	0	194

Table A.4: Confusion matrix showing the results of the Neural Network with 2 units in the hidden layer applied to the training data.

training data produced a recall of 90%, false positive rate of 1.39%, and precision of 98.9%. The three unit neural network showed improved performance over the two unit, but only when

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	71	10
	Vehicle Present	1	90

Table A.5: Confusion matrix showing the results of the Neural Network with 2 units in the hidden layer applied to the test data.

validated over the training data. Test data results were identical. Validation recall was 99.5%, false positive rate was 0%, and precision was 100%.

A.5 Parts-based Detector ROC Curve

Figure A.1 is a graph of the ROC curve for the parts-based detector on a linear scale. The x-axis is false positive rate in percent of false positive detections per windows searched, and the y-axis is the recall of vehicles in the test imagery.

		Truth	
		No Vehicle	Vehicle Present
Prediction	No Vehicle	160	1
	Vehicle Present	0	199

Table A.6: Confusion matrix showing the results of the Neural Network with 3 units in the hidden layer applied to the training data.

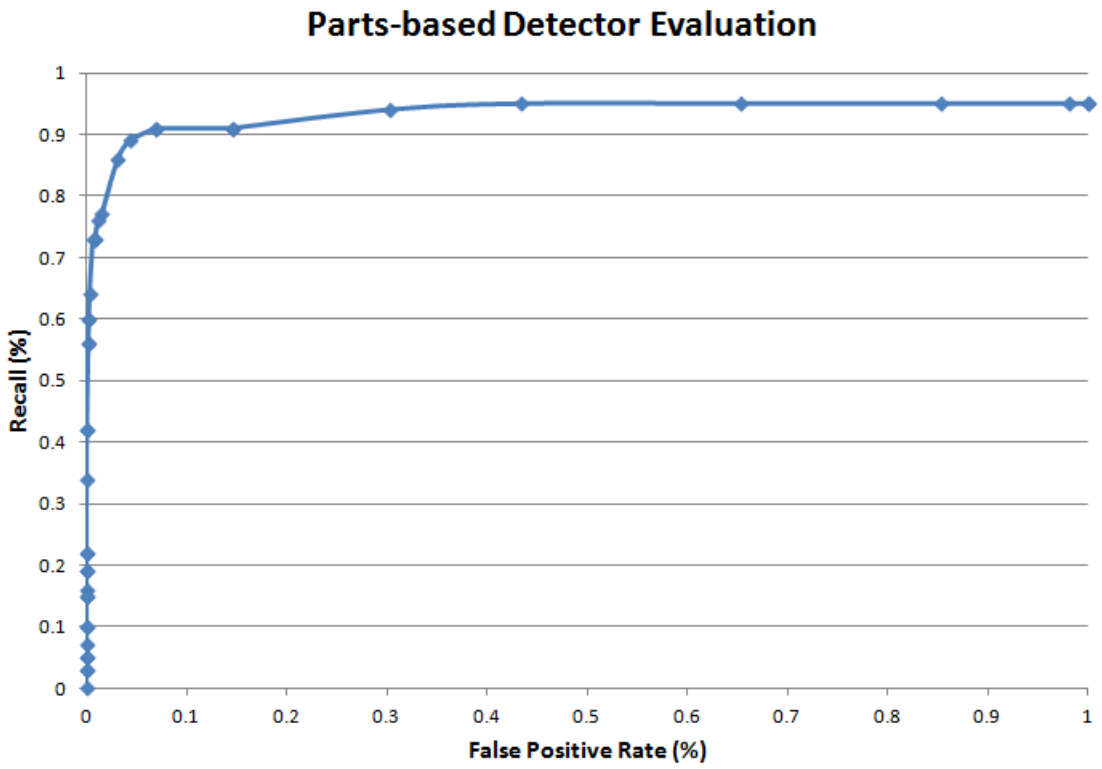

Figure A.1: The ROC curve for the parts-based detector compared to the aligned Cascade 02 detector, on a logarithmic scale.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudly Knox Library
   Naval Postgraduate School
   Monterey, California

3. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

4. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

5. Marine Corps Tactical Systems Support Activity (Attn: Operations Officer)
   Camp Pendleton, California

6. Daniel E. Cunningham, Ph.D.
   SPAWAR Systems Center Pacific
   Head, Communications Technologies Branch, Code 55210 Local Lead, Space Systems Engineering, Code 57000

7. Heidi Buck
   Head, Advanced Analysis Systems Branch (56220) Space and Naval Warfare Systems Center, Pacific

8. Director, Studies and Analysis Division, MCCDC, Code C45
   Quantico, Virginia

9. Head, Information Operations and Space Integration Branch,
   PLI/PP&O/HQMC, Washington, DC